The Double Heston Model via Filtering Methods

by

Elia N Namundjebo

Thesis presented in partial fulfilment of the requirements for the degree of Master of Science in Financial Mathematics in the Faculty of Science at Stellenbosch University

Supervisor: Prof. Jeff Sanders
Co-supervisor: Prof. Ronald Becker

December 2016

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: December 2016

Abstract

The Double Heston Model via Filtering Methods

E. N Namundjebo

Department of Mathematical Science,
Mathematics Division,
University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa.

Thesis: MSc

May 2016

Stochastic volatility models are well-known for their ability to generate a volatility smile for financial securities. The development of the stochastic volatility models followed shortly after the crash of 1987 which violates the Black-Scholes model which has constant volatility. In this study we introduce non-linear filtering methods to estimate the implied volatilities of the Double Heston model. We compare our results to the Standard Heston model. The non-linear filtering methods used are the extended Kalman filter, the unscented Kalman filter and the particle filter. We combine the filtering methods together with the maximum likelihood estimation method to estimate the model's hidden parameters. Our numerical results show that the Double Heston model fits the market implied volatilities better than the Standard Heston model. The particle filter also performs better than the other two filters.

Keywords: Stochastic volatility model, Double Heston model, non-linear filtering, maximum likelihood estimation.

JEL Classification: C11, C13, C60, G12

Uittreksel

Die dubbel Heston Model van Filter Metodes

("Die dubbel Heston Model van Filter Metodes")

E. N Namundjebo

Departement Wiskundige Wetenskappe,
Universiteit van Stellenbosch,
Privaatsak X1, Matieland 7602, Suid Afrika.

Tesis: MSc

Mei 2016

Stogastiese wisselvalligheid modelle is goed bekend vir hul vermoë om'n wisselvalligheid glimlag vir finansiële sekuriteite te genereer. Die ontwikkeling van die stogastiese wisselvalligheid modelle het gevolg kort nadat die ongeluk van 1987 wat die Black-Scholes model wat konstant wisselvalligheid oortree het. In hierdie studie stel ons nie-lineêre filter metodes voor om die geïmpliseerde wisselings in die Double Heston Model te skat. Ons vergelyk ons resultate aan die Standard Heston model. Die nie-lineêre filter metodes wat gebruik word is die uitgebreide Kalman filter, die reuklose Kalman filter en die deeltjies filter. Ons kombineer die filter metodes saam met die maksimum annneemlikheidsberaming metode om verborge parameters van die model te skat. Ons numeriese resultate dui daarop dat die Double Heston model pas die mark geïmpliseerde volatiliteit en beter as die Standard Heston model. Die deeltjie filter presteer ook beter as die ander twee filters.

Acknowledgements

I would like to thank my project supervisor, Prof Ronald Becker for his guidance and support throughout this research. Martha Kamkuemah thank you for your valuable comments. Thanks are also due to AIMS and Namfisa for funding my studies. The hospitality of AIMS is gratefully acknowledged.

Contents

Chapter 1

Introduction

Studying the implied volatility smile is important for understanding market prices and fluctuations. It has played a vital role in derivatives pricing and hedging, portfolio selection, risk management, market marking and monetary policy making, amongst others. A risk manager needs to know the likelihood that the portfolio he is holding is going to face high risk (with high volatility) in the future. An option trader will want to estimate the volatility fluctuations of the contract until maturity. An investor would like to know his stock future price movements. Policymakers rely on market volatility to determine the vulnerability of financial markets and the economy. There is vast literature in modelling the volatility smile. The models with a stochastic variance process known as stochastic volatility models are efficient in modelling the smile. However, implementing the stochastic volatility models remains a challenge. In this dissertation, we introduce non-linear filtering techniques for estimating these models as well as interpolating implied volatility term structure that precisely matches the market implied volatility smile.

The modelling of volatility fluctuations is challenging because volatility is uncertainty. The Black-Scholes (1973) model assumes a constant volatility in the option pricing model. The empirical literature on option pricing models has documented arbitrage due to the constant-volatility of the Black-Scholes option pricing model. After the work of Hull and White (1987) and Scott (1987), the majority of option pricing models can be characterized as stochastic volatility models, in which the volatility of the asset returns is driven by a stochastic variance process. A stochastic volatility model of an asset is a model of the form

$$\begin{aligned} dS_t &= rS_t dt + V_t^p dW_t \\ dV_t &= a(t, V_t)dt + b(t, V_t)dZ_t \end{aligned}$$

where r, p are positive constants and the subscript t is the time-step. The first equation defines the stock price process and the second equation defines the

volatility process. The functions for volatility a and b are deterministic, and W_t, Z_t are mutually correlated Brownian motions.

Stochastic volatility models are widely used to model the volatility smile due to their stochastic variance process. Common stochastic volatility models are the SABR model, Hull and White (1987), Scott (1987), Stein and Stein (1991), and Heston (1993) models to name a few. The Heston (1993) model is the most popular of these option pricing models because of its closed-form expression for pricing options. However, according to Duffie *et al.* (2000), the Heston (1993) model fails to capture slope and level movements of the volatility smile. The empirical studies documented that the market data displays positive as well as negative correlations, but the Heston (1993) model always yields negative correlation. Christoffersen *et al.* (2009) proposed a two-factor stochastic volatility model, named the Double Heston model which is an extension to the Heston (1993) model. The Double Heston model consists of an asset price process driven by two uncorrelated variance processes. Empirical studies show that the Double Heston model fits the implied volatility smile much better than the Standard Heston model and it displays both negative and positive correlations in the market data.

In this study we implement the Standard Heston and the Double Heston stochastic volatility models and demonstrate the differences in pricing performance. Typically, this will be done by comparing their implied volatilities term structures. However, implementing these stochastic volatility models is challenging. Most of the challenges arise because the volatility is not directly observable. It therefore needs to be extracted from observable market quotes for example, stock prices, option prices and interest rates. The variance process consists of unknown parameters, and estimating these parameters is also a challenge. We therefore do the volatilities and parameters estimation using non-linear filtering techniques combined with the maximum likelihood estimation method.

The three non-linear filtering techniques in this study are the extended Kalman filter, the unscented Kalman filter and particle filter. We use put option prices on Dow Jones Industrial Average to estimate the implied volatilities under the Standard Heston and Double Heston models. Each of the above models uses filtering methods which allow us to compare the models' and filters performances. The filtering methods are used to estimate the models' implied volatilities, whereas the maximum likelihood estimation method estimates the model's parameters. We find that the Double Heston model captures the term structure of the market implied volatilities better than the Standard Heston model. The Heston model fails to fit the data especially at shorter maturities. The extended Kalman filter performs poorly compared to the other filters. The particle filter under the Double Heston model fits the market implied volatili-

ties smile very well.

Our study contributes and relates to the active literature in modelling the implied volatility smile using stochastic volatility models. We are not the first to implement stochastic volatility models using the filtering approach, but we are the first to use filtering methods on the Double Heston model. Previous literature implement the Standard Heston model via the filtering approach using time-series of underlying returns, see Javaheri *et al.* (2003). Li (2013) uses the filtering methods on the Standard Heston model using both stock and option prices, and claims that stock prices or option prices alone does not give good fit of the implied volatilities smile and both stock and option prices are needed for better estimates. Guo *et al.* (2014) use the Nelson-Siegel term structure model to construct the implied volatility term structure under the Double Heston model using $S\&P$ 500 index call options. In the yield curves modelling it is well known that the Nelson-Siegel model is empirically successful but theoretically weak, therefore using such an approach might contribute errors in the estimated implied volatilities. Although our study relates to Rouah (2013), it differs in the approaches used to estimate the models' volatilities as well as the parameters. Our approach is simple to implement, with little computation burden and gives precise estimates that fit the market data.

The rest of the paper is organized as follows. Chapter 2 describes the filtering methods and presents their algorithms and the maximum likelihood estimation method. Chapter 3 presents the theory behind the Standard Heston and Double Heston stochastic volatility models. We also provide the state-form representations for both model. For a formal definition of the state-space representation see Section 3.3.1. The empirical evidence of the implied volatilities term structures as well the estimated parameters are provided in Chapter 4. Finally, Chapter 5 concludes the paper. All the implementations have been done in MATLAB. We developed all the codes in this study, and referenced external/additional functions. In the Appendices, we presents theory behind the Kalman filter and the MATLAB codes used in this study.

Chapter 2

Filtering Methods

In a dynamical system with a series of past and current noisy observations, filtering is used to estimate the internal states of these systems. The system's states are unobservable, but observation variables are observable. Filtering is then used to estimate the conditional probability distribution of the system's states given the observation variables. This is done in a two-step process. The first step is known as the prediction step. In the prediction step, suppose the vector x_k represents the current system's state at the current time k, and let the prediction of x_k at time k be $\hat{x}_{k|k-1}$. The state vector $\hat{x}_{k|k-1}$ is predicted using past estimated states $\hat{x}_{k-1}$ which are assumed to be known. Filtering estimates the past states $\hat{x}_{k-1}$ without conditioning on previous observations y_{k-1}. The second step is called update step. In the update step, current states $\hat{x}_{k|k}$ are estimated by combining the predicted states $\hat{x}_{k|k-1}$ with the current observations y_k. Since the observations are noisy, we seek the best estimate $\hat{x}_{k|k}$ of x_k that minimises the error $x_k - \hat{x}_{k|k}$. This is done recursively at each time step k.

In the literature, different filtering methods are proposed. A Kalman filter is one of the optimal filtering methods widely used in the field of science, engineering and finance. It is considered easy to understand with little computational burdens. In finance it is used in estimation of risk premia, optimal asset allocation, credit risk and interest rate term structure modelling, volatility estimation, and hedging under partial observation. The Kalman filter, however, is not applicable to non-linear systems. In this chapter we discuss and present some of non-linear filtering methods that are applicable to non-linear systems. We also look at the basic concepts and algorithms for these filters. These filters are the extended Kalman filter and the unscented Kalman filter. Since in general, the dynamic systems that represent the states consist of unknown parameters. Therefore, we also discuss maximum likelihood estimation method for estimating the parameters. Lastly, we present another filtering method which is slightly different from the Kalman filtering extensions called the particle filter.

We consider a discrete dynamical system with unobservable state vector x_k, for $k = 1, 2, \ldots$, where k represents time

$$x_k = f_k(x_{k-1}, w_k) \tag{2.0.1}$$

and f_k is a possibly non-linear and time-dependent function that represents the evolution of the state process x_k. The state process is driven by noise denoted by w_k.

Suppose we are also given observable vector y_k at time k

$$y_k = h_k(x_k, v_k) \tag{2.0.2}$$

where h_k is a possibly non-linear and time-dependent function that defines the measurement y_k. The observations noise is denoted by v_k. In general, the state process in Equation 2.0.1 is called *the state transition equation* and the observation process in 2.0.2 is called *measurement equation.*

To estimate the unobservable state x_k of the system at time k given all the observations up to time k, $y_{1:k}$, we can use Bayes rule to compute the conditional probability density function

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})} \tag{2.0.3}$$

where $p(\cdot)$ denotes probability density, $p(y_k|x_k)$ is the measurement probability or the likelihood function of the observation y_k given a state x_k. Also

$$p(y_k|y_{1:k-1}) = \int p(y_k|x_k)p(x_k|y_{1:k-1})dx_k$$

and

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1} \tag{2.0.4}$$

The components of Equation 2.0.3 and 2.0.4 are now explained below:

- $p(x_k|y_{1:k-1})$ means the probability density of the current state x_k conditioned on the measurements up to the timestep $k-1$. It is the probability density predicted, and the computation of this is known as the prediction step.

- $p(x_k|y_{1:k})$ means the probability density of the current state x_k conditioned on all the previous and current measurements. It is the probability density updated and the computation of $p(x_k|y_{1:k})$ is known as the update step.

The analytical expressions or numerical approximations for these probability distributions depend on the dynamic nature of the system, whether the state transition and measurement functions are linear or non-linear Gaussian or non-linear non-Gaussian. Different approximation approaches have been used, for example, the Monte Carlo Sampling. In our study we use the filtering approach. In the next sections we discuss different filtering techniques. We start with the Kalman filter. However, we are not going to use this technique for states estimation because our study focuses only on non-linear models and the Kalman filter is only optimal for linear systems.

2.1 Kalman Filter

Suppose the state function f_k from Equation 2.0.1 and the measurement function h_k from the Equation 2.0.2 are linear and their corresponding noise w_k and v_k respectively, are Gaussian and additive. Equation 2.0.1 therefore reduces to

$$x_k = M_k x_{k-1} + w_k \tag{2.1.1}$$

and Equation 2.0.2 becomes

$$y_k = H_k x_k + v_k \tag{2.1.2}$$

where we replaced the function f_k with a matrix M_k that defines the state transition evolution, and h_k with a matrix H_k that defines the measurement process and they are assumed to be known. The state noise $w_k \sim N(0, Q_k)$ and the measurement noise $v_k \sim N(0, R_k)$ are assumed to be uncorrelated Gaussian random variables (N is the n-dimensional normal distribution). Also w_k, v_k are independent of x_k, y_k respectively.

By substituting x_k and y_k from Equations 2.1.1 and 2.1.2 into the probability density functions in Equations 2.0.3 and 2.0.4, the analytical computations lead us to the Kalman filtering algorithm, see Appendix A. The distributions are normal and can be written as

$$p(x_k|x_{k-1}) \simeq N(M_k x_{k-1}, Q_k),$$

$$p(y_k|x_k) \simeq N(H_k x_k, R_k).$$

The idea is to find the estimate of the state vector x_k given the observations y_k. Using Kalman filtering, we proceed in two steps, the prediction step and the update step. In the prediction step, prediction of the state vector x_k is estimated from the previous estimated states, denoted $\hat{x}_{k-1}$. And

$$\hat{x}_{k|k-1} = M_k \hat{x}_{k-1}$$

The subscript $k|k-1$ denotes the estimated state of the state x_k using previous $(k-1)$ estimated states. We will also use a subscript $k|k$ for the estimates of

x_k using estimated states at $k|k-1$.

Note that the calculation of the previous states is done using expectation of x_k given in Equation 2.1.1.

The estimation error is given by

$$e_k^- = x_k - \hat{x}_{k|k-1}$$

and the estimate error covariance

$$P_k^- = E[e_k^- e_k^{-T}]$$

We also compute the prediction of the observations which is given by

$$\hat{y}_k = H_k \hat{x}_{k|k-1}$$

In the update step, the estimation of the current state $\hat{x}_{k|k}$ is given by the predicted states $\hat{x}_{k|k-1}$ and a measurement residual weighted by Kalman gain K_k

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_k)$$

The measurement residual is computed from $b_k = y_k - \hat{y}_k$. The estimate error is

$$e_k = x_k - \hat{x}_{k|k}$$

and the estimate error covariance

$$P_k = E[e_k e_k^T]$$

The Kalman gain K_k is an averaging factor which is the key feature of the Kalman filter. Since we already known the predicted states $\hat{x}_{k|k-1}$ and $\hat{y}_k$, then the value of the Kalman gain K_k is set so that it minimizes the variance of e_k. If $K_k = 0$, then only the predicted states are considered causing the measurements to be ignored. If $K_k = 1$, then the predicted states are ignored completely and only the measurements being considered. With a high value of K_k, the Kalman filter puts more weight on the measurements and use more measurements to minimize the errors. If the Kalman gain is low, the filter follows the predicted states more closely, removing out noise. We will always have $0 \leq K_k \leq 1$.

Below we present the standard set of recursive equations for Kalman filtering. To start the process, we first need to initialize the states x_0 and its mean square error matrix P_0.

See Appendix A for the derivation of the prediction and update equations in the Kalman filter.

Algorithm 1 Kalman filter algorithm

Step 1: Initialize

$$\hat{x}_0 = E[x_0]$$
$$P_0 = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T]$$

Step 2: Loop
for $k = 1$ to N **do**
    prediction step
    $\hat{x}_{k|k-1} = M_k \hat{x}_{k-1}$
    $P_k^- = M_k P_{k-1} M_k^T + Q_{k-1}$
    $\hat{y}_k = H_k \hat{x}_{k|k-1}$
    $F_k = H_k P_k^- H_k^T + R_k$
    Update step
    $b_k = y_k - \hat{y}_k$
    $K_k = P_k^- H_k^T (F_k)^{-1}$
    $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k b_k$
    $P_k = P_k^- - K_k H_k P_k^-$
end for

For parameter estimation, suppose Ω is the set of the unknown parameters in the linear model represented by Equations 2.1.1 and 2.1.2. Then a Maximum Likelihood Estimator can be used to estimate the parameters. Therefore, we need to maximize the likelihood function,

$$L(y_1, \ldots, y_k; \Omega) = \prod_{k=1}^{N} p(y_k | y_{1:k-1}; \Omega) \tag{2.1.3}$$

with $p(\cdot)$ denoting a multivariate density function. If the forecasting errors in the problem are Gaussian, then the multivariate Gaussian density function is

$$L(y_{1:k}) = \frac{1}{(2\pi)^{d/2}\sqrt{\det(F_k)}} \exp\left(-\frac{(y_k - \hat{y}_k)^T F_k^{-1} (y_k - \hat{y}_k)}{2}\right)$$

where F_k is defined in algorithm 1 and d is dimension of y_k. For computational and theoretical reasons, we usually work with the log-likelihood. We express Equation 2.1.3 as a log-likelihood function

$$\log L(y_1, \ldots, y_k; \Omega) = \sum_{t=1}^{k} \left(-\frac{d}{2}\log(2\pi) - \frac{1}{2}\log|F_k^{-1}| - \frac{1}{2}(y_k - \hat{y}_k)^T F_k^{-1} (y_k - \hat{y}_k)\right) \tag{2.1.4}$$

To obtain the parameters, we minimize the above function over the set of parameters Ω using a numerical optimization routine for example, the Nelder-Mead optimization.

However, some systems can be more complex and non-linear, where the non-linearity can be in the states process or in the measurements process or both. Kalman filter is only optimal for linear systems. Therefore the need for non-linear filters. Next we discuss an extension of the Kalman filter known as the extended Kalman filter which can handle non-linear Gaussian systems.

2.2 Extended Kalman Filter

Extended Kalman filter is another filtering technique that is widely used in the mathematical modeling. It is an extension to the optimal Kalman filter and it is applicable to non-linear dynamical systems.

Suppose the state transition function f_k and the observation function h_k given in Equations 2.0.1 and 2.0.2 respectively are both non-linear and their corresponding noises are uncorrelated Gaussian random variables, where w_k has zero mean and covariance Q_k and v_k has zero mean and covariance R_k. If the conditional densities in 2.0.3 and 2.0.4 are Gaussian, then the extended Kalman filter can be used to estimate the state vector x_k given the observations y_k at time step k.

Similarly to the Kalman filter, the extended Kalman filter algorithm is also grouped into two steps, the prediction step and the update step. In the prediction step, the states are predicted as

$$\hat{x}_{k|k-1} = f_k(\hat{x}_{k-1}, 0)$$

The covariance is computed from the linearization of the non-linear functions in the state transition and measurement equations. The linearization of the non-linear state and measurement functions is defined by the Jacobian matrices:

$$A_{ij} = \frac{\partial f_i(\hat{x}_{k-1}, 0)}{\partial x_j}, \qquad W_{ij} = \frac{\partial f_i(\hat{x}_{k-1}, 0)}{\partial w_j}$$
$$H_{ij} = \frac{\partial h_i(\hat{x}_{k|k-1}, 0)}{\partial x_j}, \qquad U_{ij} = \frac{\partial h_i(\hat{x}_{k|k-1}, 0)}{\partial v_j}$$

So that the predicted state covariance is given by

$$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T$$

Prediction of the measurement is given by

$$\hat{y}_k = h_k(\hat{x}_{k|k-1}, 0)$$

with covariance

$$F_k = H_k P_k^- H_k^T + U_k R_k U_k^T$$

In the update step, the state vector x_k is estimated using the predicted states $\hat{x}_{k|k-1}$ and the measurement residual weighted by the Kalman gain K_k.

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k b_k$$

where $b_k = y_k - \hat{y}_k$ is the measurement residual.

The optimal gain is

$$K_k = P_k^- H_k^T (F_k)^{-1}$$

and the updated covariance

$$P_k = P_k^- - K_k H_k P_k^-$$

These steps complete the extended Kalman filter algorithm.

Below we present the extended Kalman filter algorithm

Algorithm 2 Extended Kalman filter algorithm

Step 1: Initialize

$$\hat{x}_0 = E[x_0]$$
$$P_0 = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T]$$

Step 2: Loop
for $k = 1$ to N **do**
    prediction step
    $\hat{x}_{k|k-1} = f_k(\hat{x}_{k-1}, 0)$
    $P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T$
    $\hat{y}_k = h_k(\hat{x}_{k|k-1})$
    $F_k = H_k P_k^- H_k^T + U_k R_k U_k^T$
    Update step
    $b_k = y_k - \hat{y}_k$
    $K_k = P_k^- H_k^T (F_k)^{-1}$
    $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k b_k$
    $P_k = P_k^- - K_k H_k P_k^-$
end for

If the state transition noise w_k and the measurement noise v_k are additive such that

$$x_k = f_k(x_{k-1}) + w_k$$
$$y_k = h_k(x_k) + v_k$$

then the Jacobian matrix of f_k with respect to the system noise W_k and the Jacobian matrix of h_k with respect to the measurement noise U_k are not necessary. The predicted covariance becomes

$$P_k^- = A_k P_{k-1} A_k^T + Q_{k-1}$$

and the residual covariance

$$F_k = H_k P_k^- H_k^T + R_k$$

2.3 Unscented Kalman Filter

The linearization for the non-linear functions (the state transition and measurement equations) in the extended Kalman filter has been criticized. It is argued that for highly non-linear systems, the extended Kalman filter has provided poor estimates. Since filtering highly depends on the Kalman gain which is calculated from the covariance of the states and measurements, then a poor estimate of the covariance gives bad values of the Kalman gain. The computation of the covariance in the extended Kalman filter depends on the linearized non-linear functions, and a poor representation of these functions gives poor state estimate due to the poor calculation of covariance. Julier and Uhlmann (1996) proposed a filtering method called unscented Kalman filter. They argued that the unscented Kalman filter estimates highly non-linear systems with Gaussian distributions more accurate than the extended Kalman filter. This is because the unscented Kalman filter approximates the states covariance better than the extended Kalman filter. Julier and Uhlmann (1996) also argued that the extended Kalman filter is difficult to implement, because it requires approximation methods for computation of the Jacobian matrices.

Unlike the extended Kalman filter, the unscented Kalman filter does not approximate the non-linear process and observation models, it uses the true non-linear models and rather approximates the distribution of the state random variable (Van Der Merwe *et al.* (2000)). The distribution of the states in Equation 2.0.3 is approximated by a set of well chosen deterministic sample points. These sample points are called sigma points. Each sigma point is associated with two weights. The sigma points completely capture the true mean and covariance of the states. Below we give an example of how to generate the sigma points and their corresponding weights.

Suppose

$$y = q(x), \tag{2.3.1}$$

where q is a non-linear function. To compute the probability density function of y given the probability density function of x which is a normal distribution,

we proceed as follows:

Suppose $\dim(x) = L$ and x has mean $\hat{x}$ and covariance matrix P_x. Then we generate a set of $2L+1$ weighted sigma points $\{W_i, X(i)\}$ such that

$$\begin{aligned} X(0) &= \hat{x} \\ X(i = 1, \ldots, L) &= \hat{x} + \left(\sqrt{(L+\lambda)P_x}\right)_i \\ X(i = L+1, \ldots, 2L) &= \hat{x} - \left(\sqrt{(L+\lambda)P_x}\right)_{i-L} \end{aligned}$$

where λ is a scaling parameter, defined by

$$\lambda = \alpha^2(L + \kappa) - L$$

and α determines the spreads of sigma points around x, and usually set as small as possible (e.g. $10^{-4} \leq \alpha \leq 1$). A secondary scaling parameter, κ, is usually set to $\kappa = 3 - L$.

Each sigma point $X(i)$ for $i = 1, \ldots, 2L$ is associated with a set of two weights $W_i^{(m)}, W_i^{(c)}$ defined as

$$\begin{aligned} W_0^{(m)} &= \frac{\lambda}{L+\lambda} & W_0^{(c)} &= \frac{\lambda}{L+\lambda} + 1 - \alpha^2 + \beta \\ W_i^{(m)} &= \frac{1}{2(L+\lambda)} & W_i^{(c)} &= \frac{1}{2(L+\lambda)} \end{aligned} \tag{2.3.2}$$

where β is used to incorporate prior knowledge of the distribution of x, and $\beta = 2$ is optimal for Gaussian distributions.

We then propagate the sigma points into the non-linear function q, such that

$$Y(i = 0, \ldots, 2L) = q(X(i))$$

Then the mean of y is given by

$$\hat{y} = \sum_{i=0}^{2L} W_i^{(m)} Y(i)$$

with covariance

$$P = \sum_{i=0}^{2L} W_i^{(c)} \left(Y(i) - \hat{y}\right)\left(Y(i) - \hat{y}\right)^T.$$

Now to compute the probability density function in 2.0.3 and 2.0.4 using the unscented Kalman filter, we first generate the sigma points of the state transition equation in 2.0.1. The sigma points can be generated as was done above. Next, we discuss the algorithm for the unscented Kalman filter as explained in Wan and Van Der Merwe (2000). We start with the generation of the sigma points and the propagation of these sigma points into the non-linear state transition and measurement process.

Suppose we have a non-linear state transition Equation 2.0.1 and a non-linear measurement Equation 2.0.2. We define dimensions of the state process and the state noise

$$N_x = \dim(x),\ N_w = \dim(w)$$

and dimensions of the measurement process and the measurement noise

$$N_y = \dim(y),\ N_v = \dim(v)$$

respectively.

Put $L = N_x + N_w + N_v$, we then construct an L-dimensional column vector (x_k^a) whose entries are the state process, and the state and measurement noise:

$$x_k^a = [x_k, w_k, v_k]^T \tag{2.3.3}$$

Assume x_k^a of dimension L has mean $\bar{x}$ and covariance matrix P_x. We can construct an $L \times (2L+1)$-matrix

$$\chi^a = [\chi_0, \chi_1, \ldots, \chi_{2L}]$$

of sigma points, with columns defined by

$$\begin{aligned}
\chi_0 &= \bar{x} \\
\chi_i &= \bar{x} + \left(\sqrt{(L+\lambda)P_x}\right)_i \quad \text{for} \quad i = 1, \ldots, L \\
\chi_i &= \bar{x} - \left(\sqrt{(L+\lambda)P_x}\right)_{i-L} \quad \text{for} \quad i = L+1, \ldots, 2L
\end{aligned}$$

where $\lambda, \alpha, \kappa, \beta, W_i^{(m)}$ and $W_i^{(c)}$ (for $i = 0, \ldots, 2L$) are as defined above in Equation 2.3.2.

The matrix χ^a of sigma points obtained can be decomposed into:

$$\chi^a = \begin{bmatrix} \chi^x \\ \chi^w \\ \chi^v \end{bmatrix}$$

where χ^x is $N_x \times (2L+1)$-dimensional, χ^w is $N_w \times (2L+1)$-dimensional, and χ^v is $N_v \times (2L+1)$-dimensional.

After calculating the sigma points, we then propagate them through the state transition process defined in Equation 2.0.1. Next we present step-by-step unscented Kalman filter algorithm.

Similarly to the extended Kalman filter, the unscented Kalman filter is also grouped into two steps, the prediction and update step. We start with an initial choice for the state vector $\hat{x}_0$.

2.4 Maximum Likelihood Estimation(MLE)

The state transition process in Equation 2.0.1 consists of unknown parameters. Let Ω be a set of all unknown parameters in 2.0.1. We can rewrite Equation 2.0.1 as

$$x_k = f_k(x_{k-1}, w_k; \Omega)$$

Therefore a calibration method is required to estimate the set of hidden parameters Ω. There exists a rich literature on parameter estimation methods. In this study we follow the Javaheri *et al.* (2003) approach of maximum likelihood estimation method (MLE). We need to maximize the likelihood function

$$p(y_1, \ldots, y_k; \Omega) = \prod_{k=1}^{N} p(y_k | y_{1:k-1}; \Omega)$$

over the parameter set Ω. If the forecasting errors in the problem are Gaussian, then the multivariate Gaussian density function is

$$p(y_1, \ldots, y_k; \Omega) = \frac{1}{(2\pi)^{N_y/2}\sqrt{\det(P_{y_k y_k})}} \exp\left(-\frac{b_k^T P_{y_k y_k}^{-1} b_k}{2}\right)$$

where $N_y = \dim(y)$ and $b_k = y_k - \hat{y}_k$.

Instead we can minimise minus the log of this likelihood function, so that

$$L_{1:N_y} = \frac{1}{2}\sum_{k=1}^{N_y}\left(N_y \log(2\pi) + \log(\det(P_{y_k y_k}^{-1})) + b_k^T P_{y_k y_k}^{-1} b_k\right) \tag{2.4.1}$$

where $L_{1:N_y} = -\log p(y_1, \ldots, y_k; \Omega)$.

The filtering techniques already provided us with the coefficients in $L_{1:N_y}$ such that $P_{y_k y_k}$, y_k, $\hat{y}_k$ are defined under the unscented Kalman filter algorithm.

For the extended Kalman filter,

$$\hat{y}_k = h_k(\hat{x}_{k|k-1}, 0)$$

Algorithm 3 Unscented Kalman filter algorithm

1: Step 1: Initialize

$$\hat{x}_0 = E[x_0]$$
$$P_0 = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T]$$
$$\hat{x}_0^a = E[x_0^a] = \begin{bmatrix} \hat{x}_0 \\ 0 \\ 0 \end{bmatrix}$$
$$P_0^a = E[(x_0^a - \hat{x}_0^a)(x_0^a - \hat{x}_0^a)^T] = \begin{bmatrix} P_0 & 0 & 0 \\ 0 & P_w & 0 \\ 0 & 0 & P_v \end{bmatrix}$$

where P_w and P_v are the covariance of the state noise and measurement noise respectively.

2: Step 2: Loop

3: **for** $k = 1$ to N_y **do**

4: Sigma points:

$$\chi_{k-1}^a(0) = \hat{x}_{k-1}^a$$

for $i = 1, \ldots, L$

$$\chi_{k-1}^a(i) = \hat{x}_{k-1}^a + \left(\sqrt{(L+\lambda)P_{k-1}^a}\right)_i$$

and for $i = L+1, \ldots, 2L$

$$\chi_{k-1}^a(i) = \hat{x}_{k-1}^a - \left(\sqrt{(L+\lambda)P_{k-1}^a}\right)_{i-L}$$

where the subscripts i and $i - L$ correspond to the i^{th} and $i - L^{th}$ columns of the square-root matrix.

5: Since χ_{k-1}^a is known, we also know $\chi_{k-1}^x, \chi_{k-1}^w, \chi_{k-1}^v$:

$$\chi_{k-1}^a = \begin{bmatrix} \chi_{k-1}^x \\ \chi_{k-1}^w \\ \chi_{k-1}^v \end{bmatrix}$$

6: *prediction step*

7: Sigma points of x_k:

$$\chi_{k|k-1}(i) = f(\chi_{k-1}^x(i), \chi_{k-1}^w(i)) \qquad \text{for} \quad i = 0, \ldots, 2L.$$

8:

$$\hat{x}_{k|k-1} = \sum_{i=0}^{2L} W_i^{(m)} \chi_{k|k-1}(i)$$

9:

$$P_k^- = \sum_{i=0}^{2L} W_i^{(c)} \left(\chi_{k|k-1}(i) - \hat{x}_{k|k-1}\right)\left(\chi_{k|k-1}(i) - \hat{x}_{k|k-1}\right)^T$$

10: Sigma points of y_k

$$\mathcal{Y}_{k|k-1}(i) = h(\chi_{k|k-1}(i), \chi_{k-1}^v(i))$$

Algorithm 3 My algorithm (continued)

11:
$$\hat{y}_k = \sum_{i=0}^{2L} W_i^{(m)}(i)\mathcal{Y}_{k|k-1}(i)$$

12:
$$P_{y_k y_k} = \sum_{i=0}^{2L} W_i^{(c)} \left(\mathcal{Y}_{k|k-1}(i) - \hat{y}_k\right)\left(\mathcal{Y}_{k|k-1}(i) - \hat{y}_k\right)^T$$

13: Joint covariance of x_k and y_k is given by

$$P_{x_k y_k} = \sum_{i=0}^{2L} W_i^{(c)} \left(\chi_{k|k-1}(i) - \hat{x}_{k|k-1}\right)\left(\mathcal{Y}_{k|k-1}(i) - \hat{y}_k\right)^T$$

14: *Update step*

15: The Kalman gain is given by

$$K_k = P_{x_k y_k} P_{y_k y_k}^{-1}$$

16: The measurement residual

$$b_k = y_k - \hat{y}_k$$

17: The new state estimate

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k b_k$$

18: $P_k = P_k^- - K_k P_{y_k y_k} K_k^T$

19: Then ultimately, set

$$\hat{x}_k^a = E[x_k^a] = \begin{bmatrix} \hat{x}_{k|k} \\ 0 \\ 0 \end{bmatrix}$$

$$P_k^a = E[(x_k^a - \hat{x}_k^a)(x_k^a - \hat{x}_k^a)^T] = \begin{bmatrix} P_k & 0 & 0 \\ 0 & P_w & 0 \\ 0 & 0 & P_v \end{bmatrix}$$

20: **end for**

and

$$P_{y_k y_k} = H_k P_{k|k-1} H_k^T + U_k R_k U_k^T$$

For further explanations on MLE for the extended and unscented Kalman filters, see Javaheri (2011) (pg 67-68).

The minimization of $L_{1:N}$ over the set of parameters Ω could be done via a numerical optimization routine such as the Nelder-Mead algorithm (Lagarias *et al.* (1998)), Secant-Levenberg-Marquardt method, the modified Sequential Quadratic Programming (modSQP) method or the Simulated Annealing method, (see Kienitz and Wetterau (2012), Chapter 9). To obtain parameters, this study uses MATLAB's "off-the-shelf" optimizer `fminsearch` and the modSQP method.

2.5 Particle Filter

Another popular filtering technique is the Sequential Monte Carlo method known as the particle filter. This method is widely used in fluid mechanics. When a non-linear system is non-Gaussian then extended and unscented Kalman filters are not optimal methods to use. The particle filter method uses Monte Carlo simulation to compute the posterior density function given in Equation 2.0.3. Monte Carlo simulation is a well known approximation method which uses a set of well chosen random numbers. These random samples are then used to approximate state distributions by performing many iterations, where each iteration uses a different set of random values. Similarly to the Monte Carlo, the particle filter approximate the state distributions with a finite set of weighted random samples drawn from a known, easy to sample, proposal distribution ($q(x_{0:k}|y_{1:t})$). These random samples are called particles and at time step k we might denote n particles for the state x_k as

$$x_k^{(1)}, x_k^{(2)}, \ldots, x_k^{(n)}.$$

The particles $x_{0:k}^{(i)}$ for $i = 1, \ldots, n$ are independent and identically distributed.

Each particle is assigned an importance weight (r_k) which determines its probability of being sampled from the proposal distribution. The weighted set of n particles at time step k will be denoted as $\{x_k^{(i)}, r_k^{(i)}\}$ for all $i = 1, \ldots, n$.

The posterior density function from Equation 2.0.3 can be approximated as follows

$$p(x_{0:k}|y_{1:k}) := \sum_{i=1}^{n} r_k^{(i)} \delta\left(x_{0:k} - x_{0:k}^{(i)}\right) \tag{2.5.1}$$

where $\delta(\cdot)$ is a delta function. All we need to do to evaluate the transition probability in Equation 2.5.1, we need to generate a set of particles from a

proposal distribution and iteratively compute the importance weights. This is grouped into three steps:

1. Sampling
2. Computing the particle weights
3. Resampling.

The next subsections discuss the above steps in detail.

2.5.1 Sampling and computing the weights

The challenging part in implementing the particle filter, is the choice of the proposal distribution $q(\cdot)$ for generating particles. A proposal distribution should contain the outputs of the posterior probability distribution and generation of the samples should be done randomly. This has been a critical design issue and several proposal distributions are proposed in the literature. Lu *et al.* (2015) discuss different proposal distributions for the particle filter and which are better suited to a particular target distribution. Different proposal functions based on the Kalman filter are proposed by Van Der Merwe *et al.* (2000). In this study we use a proposal function by Doucet *et al.* (2000) given as

$$q(x_k|x_{0:k-1}, y_{1:k}) = p(x_k|x_{k-1})$$

For each particle $x_k^{(i)}$ (for $i = 1, \ldots, n$) drawn from $q(x_k|x_{0:k-1}^{(i)}, y_{1:k})$ has an importance weight defined by:

$$\begin{aligned} r_k^{(i)} &= r_{k-1}^{(i)} \frac{p(y_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{q(x_k^{(i)}|x_{0:k-1}^{(i)}, y_{1:k})} \\ &= r_{k-1}^{(i)} \frac{p(y_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{p(x_k^{(i)}|x_{k-1}^{(i)})} \\ &= r_{k-1}^{(i)} p(y_k|x_k^{(i)}) \end{aligned} \tag{2.5.2}$$

The importance weights are normalized, so that all the weights sum to 1,

$$\tilde{r}_k^{(i)} = r_k^{(i)} \left[\sum_{j=1}^{n} r_k^{(j)} \right]^{-1}.$$

For further explanations on the particle filter, see Arulampalam *et al.* (2002), Van Der Merwe *et al.* (2000), Thrun (2002), and Doucet *et al.* (2000).

2.5.2 Resampling

One of the major problems with particle filtering is particle degeneration. This means that after several iterative procedures, a few importance weights become very large and all the other particles will have very small importance weights to the point that they become negligible. This has a harmful effect on the accuracy of the estimates, since a large number of particles (those with low weights) are removed from the sample set. Degeneration of the particles has an effect on computational time, since time will be wasted on low-weighted particles that are no longer useful in the estimation. Resampling is a strategy proposed to reduce degeneration of particles.

In resampling, all the particles with low importance weights are eliminated and a new set of n equally weighted particles is drawn from the remained particles. A common way to measure the degeneracy is by an estimate of the effective sample size given as follows

$$N_{eff} = \frac{1}{\sum_{i=1}^{n}(r_k^i)^2}.$$

The resampling step is taken when $N_{eff} < n_e$, where n_e is usually set as $\frac{n}{2}$. At the end of the resampling procedure all the importance weights of the new particles will be equal to $1/n$. The importance weights are determined as follows

$$r_k^{(i)} = \tilde{r}_k^{(i)} = \frac{1}{n}$$

For more details on resampling algorithms see Hol *et al.* (2006), Arulampalam *et al.* (2002) and Van Der Merwe *et al.* (2000).

2.5.3 The particle filter algorithm

From the previous subsection, we have explained how to generate particles from a proposal distribution. We also discussed how to compute their corresponding importance weights. To avoid degeneracy of particles, resampling of particles was also discussed. We now present algorithm of the particle filter as outlined in Van Der Merwe *et al.* (2000).

Algorithm 4 Particle filter algorithm

Initialize
for $k = 0$ **do**
    For $i = 1, \ldots, n$, draw the state's particles $x_0^{(i)}$ from $p(x_0)$
end for
for $k = 1$ to N **do**
    (a) Sampling step
    For $i = 1, \ldots, n$, sample $x_k^{(i)}$ from $q(x_k|x_{0:k-1}^{(i)}, y_{1:k})$
    For $i = 1, \ldots, n$, compute the importance weights

$$r_k^{(i)} = r_{k-1}^{(i)} \frac{p(y_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{q(x_k^{(i)}|x_{0:k-1}^{(i)}, y_{1:k})}$$

    For $i = 1, \ldots, n$, normalize the importance weights

$$\tilde{r}_k^{(i)} = r_k^{(i)} \left[\sum_{j=1}^{n} r_k^{(j)} \right]^{-1}$$

    (b) Resampling step
    if $N_{eff} < n_e$ **then**
        Eliminate particles $x_k^{(i)}$ with low importance weights $\tilde{r}_k^{(i)}$
        Draw a new set of n equally weighted particles $x_k^{(i)}$ approximately
        distributed according to $q(x_k^{(i)}|x_{k-1}^{(i)}, y_{1:k})$
        For $i = 1, \ldots, n$ set $r_k^{(i)} = \tilde{r}_k^{(i)} = \frac{1}{n}$
    else
        Do nothing
    end if
    (c) output
    Approximated posterior distribution

$$\hat{x}_{k|k} = \sum_{i=1}^{n} \tilde{r}_k^{(i)} x_k^{(i)}$$

end for

2.5.4 Maximum Likelihood Estimation for particle filter

For parameter estimation under the particle filter, we use the MLE method. Given a likelihood function at time step k

$$\begin{aligned} l_k = p(y_k|y_{1:k-1}) &= \int p(y_k|x_k)p(x_k|y_{1:k-1})dx_k \\ &= \int p(y_k|x_k)\frac{p(x_k|y_{1:k-1})}{q(x_k|x_{k-1}y_{1:k})}q(x_k|x_{k-1}y_{1:k})dx_k \\ &= \sum_{i=1}^{n} \frac{p(y_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{q(x_k^{(i)}|x_{k-1}^{(i)})} \end{aligned}$$

the log-likelihood to be maximized is

$$\ln(L_{1:n}) = \sum_{k=1}^{n} \ln(l_k) \tag{2.5.3}$$

and hence the parameters set Ω will be obtained using the same optimization methods as discussed in Section 2.4.

For further explanation of MLE under particle filtering, refer to Javaheri *et al.* (2003) and Javaheri (2011) (Page 102).

Chapter 3

Stochastic Volatility Models

The stock market crash of October 1987 caused investors to criticize the mathematical models on their ability to price options. Options are perceived to be complex derivatives due to the financial crisis. The widely used Black-Scholes (1973) model assumes that the underlying volatility is constant over the life of the derivative. Empirical studies have shown that the Black-Scholes constant volatility does not hold in equity markets. Instead of getting horizontal graphs of volatility vs maturity or volatility vs strike, various asset returns have exhibited a nonlinear behaviour, sometimes like an upward (parabolic) smile. This has become known as the volatility smile. Half of a smile is referred to as the volatility skew or volatility smirk. The quoted market prices for out-the-money put prices (and in-the-money call prices) are higher than the Black-Scholes prices (Christoffersen *et al.* (2009)). Therefore, the Black-Scholes model does not adequately capture all the features observed in the option market. To overcome this problem, recent studies assume return volatility to be time-varying and predictable. Models with time-varying volatility which are driven by their own stochastic processes are known as stochastic volatility models.

Some of the widely used stochastic volatility models for pricing options are Hull and White (1987), Bates (1996), Heston (1993), Stein and Stein (1991) and Scott (1987). The Heston (1993) model is one of the most popular stochastic volatility models for pricing equity options. The choice of the Heston model is motivated by its closed-form valuation formula that can be used to price options. Risky asset returns that follow a normal distribution cannot fully explain some features such as the smile or skew of the implied volatilities extracted from option prices. Under the Heston model, the underlying asset returns exhibit a fatter tail distribution than that of a normal distribution. Hence, the Heston model is capable of generating smile or skew of the implied volatilities. Christoffersen *et al.* (2009) proposed a two-factor model called the Double Heston model. They argue that the Standard Heston (1993) model does not always capture the dynamics of the term structure of implied volatil-

ity very well, especially at short maturities. In the Double Heston model, an asset return is driven by two-factor stochastic volatility. This has the advantage of improving the model's flexibility in modelling the volatility term structure.

In this chapter, we describe the Standard Heston model and its extension, the Double Heston model, in detail and present their characteristic functions, which are important in option valuations. We also present the state-space representations for these models, which we use in the filtering methods to estimate the volatilities.

3.1 The Heston Model

In this section, we first present the dynamic system for the Heston model under a risk-neutral measure $\mathbb{Q}$. Then lastly, we show how to price options under the Heston model.

Under a risk-neutral measure $\mathbb{Q}$, the Heston (1993) model assumes that an underlying stock price, S_t has a stochastic variance, V_t, that follows a Cox, Ingersoll and Ross (1985) process. This process is represented by the following dynamical system:

$$dS_t = (r-q)S_t dt + \sqrt{V_t} S_t dW_t \tag{3.1.1}$$

$$dV_t = \kappa(\theta - V_t)dt + \sigma\sqrt{V_t} dZ_t, \tag{3.1.2}$$

where r is a constant risk-free interest rate, q is a constant dividend, κ is a mean reversion rate for the variance. The model mean reversion level for the variance is denoted by θ and σ is a volatility of the variance. All the parameters κ, θ, σ are positive constant. The two independent Brownian motions W_k and Z_k are correlated with a constant correlation $\rho \in [-1, 1]$.

For option valuation, we follow the Albrecher *et al.* (2006) approach, such that the characteristic function of log returns $x_k = \ln(S_k/S_{k-1})$ (for $k \leq t$) of the Heston model is derived using the so called the little Heston trap. This characteristic function is only slightly different from the original formulation of Heston (1993), but it provides a better computation of the numerical integration. Heston (1993) provided the European call option closed-form solution given by

$$C(S, V, K, \tau) = S_k e^{-q\tau} P_1 - K e^{-r\tau} P_2 \tag{3.1.3}$$

where K is the strike price, and probabilities

$$P_j = \frac{1}{2} + \frac{1}{\pi} \int_0^\infty \text{Re}\left[\frac{e^{-i\phi \ln K} f_j(\phi; x_k, V_k)}{i\phi}\right] d\phi$$

for $j = 1, 2$.

The characteristic functions $f_j(\phi; x_k, V_k)$ in the probabilities are given by

$$f_j(\phi, \tau, x_k, V_k) = e^{i\phi x_k + A_j(\phi,\tau) + B_j(\phi,\tau)V_k}$$

where

$$B_j(\tau, \phi) = \frac{b_j - \rho\sigma\phi i + d_j}{\sigma^2}\left[\frac{1 - e^{d_j\tau}}{1 - g_j e^{d_j\tau}}\right],$$

$$A_j(\tau, \phi) = r\phi i\tau + \frac{a}{\sigma^2}\left[(b_j - \rho\sigma\phi i + d_j)\tau - 2\ln\left(\frac{1 - g_j e^{d_j\tau}}{1 - g_j}\right)\right],$$

$$g_j = \frac{b_j - \rho\sigma\phi i + d_j}{b_j - \rho\sigma\phi i - d_j},$$

$$d_j = \sqrt{(\rho\sigma\phi i - b_j)^2 - \sigma^2(2u_j\phi i - \phi^2)},$$

and $i = \sqrt{-1}$, $\tau = T - k$, $u_1 = \frac{1}{2}$, $u_2 = -\frac{1}{2}$, $a = \kappa\theta$, $b_1 = \kappa - \rho\sigma$, $b_2 = \kappa$ and ϕ is called the integration variable or node.

The European put options can be obtained via put-call parity. A number of papers have documented the derivations for the options premium under the Heston (1993) model as well the characteristic function see Heston (1993), Albrecher *et al.* (2006), (Rouah (2013), Chapter 1) and (Zhu (2009), Chapter 3) for further discussions.

3.2 The Double Heston Model

The Double Heston model proposed by Christoffersen *et al.* (2009), assumes that the underlying stock price, S_t is driven by two independent factors of volatility, V_t^1 and V_t^2. Under a risk-neutral framework its dynamical system is defined as follow:

$$\begin{aligned} dS_t &= (r - q)S_t dt + \sqrt{V_t^1}S_t dW_t^1 + \sqrt{V_t^2}S_t dW_t^2, \\ dV_t^1 &= \kappa_1(\theta_1 - V_t^1)dt + \sigma_1\sqrt{V_t^1}dZ_t^1, \\ dV_t^2 &= \kappa_2(\theta_2 - V_t^2)dt + \sigma_2\sqrt{V_t^2}dZ_t^2, \end{aligned} \tag{3.2.1}$$

where r is a constant risk-free interest rate, q is a constant dividend-yield and other parameters also constant. The Brownian motions W_t^1, Z_t^1 and W_t^2, Z_t^2 are correlated

$$\begin{aligned} d[W^i, Z^j]_t &= \rho_i dt \quad \text{for all} \quad i = j \\ d[W^i, Z^j]_t &= 0 \quad \text{for all} \quad i \neq j \end{aligned}$$

for $i, j = 1, 2$. Note that the constant correlation parameters $\rho_1, \rho_2 \in [-1, 1]$.

To determine the characteristic function for the Double Heston model, we first state the multi-dimensional Feynman-Kac Theorem.

Theorem 3.1. *Multi-dimensional Feynman-Kac Theorem*

Let x_k be an $n-$dimensional stochastic process with dynamics

$$dx_k = \mu(k, x_k)dk + \sigma(k, x_k)dW_k \tag{3.2.2}$$

where $k \leq t \leq T$,

- *a column vector valued function $\mu(k, x^1, \ldots, x^n) : \mathbb{R}^+ \times \mathbb{R}^n \to \mathbb{R}^n$*
- *a matrix valued function $\sigma(k, x^1, \ldots, x^n) : \mathbb{R}^+ \times \mathbb{R}^n \to \mathbb{R}^{n \times d}$*
- *W_k is a $d-$dimensional Brownian motion with independent components*

The infinitesimal generator of the process in Equation 3.2.2 is defined by

$$\mathcal{A} = \sum_{i=1}^{n} \mu_i(k, x^1, \ldots, x^n) \frac{\partial}{\partial x^i} + \frac{1}{2} \sum_{i,j=1}^{n} C_{ij} \frac{\partial^2}{\partial x^i \partial x^j} \tag{3.2.3}$$

where $C_{ij} = (\sigma\sigma^T)_{ij}$.

Let $f : \mathbb{R}^n \to \mathbb{R}^n$ be a solution of the boundary value problem

$$\begin{aligned} &\frac{\partial f}{\partial t}(t, x^1, \ldots, x^n) + \mathcal{A}f(t, x^1, \ldots, x^n) - r(t, x^1, \ldots, x^n) f(t, x^1, \ldots, x^n) = 0 \\ &f(T, x^1, \ldots, x^n) = \Phi(x^1, \ldots, x^n) \end{aligned} \tag{3.2.4}$$

for a real valued functions $r(k, x^1, \ldots, x^n) : \mathbb{R}^+ \times \mathbb{R}^n \to \mathbb{R}$ and $\Phi(x^1, \ldots, x^n) : \mathbb{R}^n \to \mathbb{R}$.

Then the solution $f(t, x^1, \ldots, x^n)$ is given by the expectation

$$f(t, x^1, \ldots, x^n) = \mathbb{E}\left[exp\left(-\int_t^T r(k, x_k)dk \right) \Phi(x_T) \right].$$

According to the Feynman-Kac Theorem 3.1, we know that f satisfies the PDF

$$\frac{\partial f}{\partial t} + \mathcal{A}f - rf = 0 \tag{3.2.5}$$

From Equation 3.2.1 and Ito's lemma, the log returns $x_k = \ln(S_k/S_{k-1})$ (in this case x_k is a scalar) are given by

$$dx_k = \left(r - q - \frac{1}{2}(V_k^1 + V_k^2)\right) dk + \sqrt{V_k^1} dW_k^1 + \sqrt{V_k^2} dW_k^2$$

This implies that the dynamical system of the Double Heston model can be written as

$$\begin{pmatrix} dx_k \\ dV_k^1 \\ dV_k^2 \end{pmatrix} = \begin{pmatrix} \left(r - q - \frac{1}{2}(V_k^1 + V_k^2)\right) dk + \sqrt{V_k^1} dW_k^1 + \sqrt{V_k^2} dW_k^2 \\ \kappa_1(\theta_1 - V_k^1)dk + \sigma_1 \sqrt{V_k^1} dZ_k^1 \\ \kappa_2(\theta_2 - V_k^2)dk + \sigma_2 \sqrt{V_k^2} dZ_k^2 \end{pmatrix}$$

If we set

$$Z_1 = \rho_1 W_1 + \sqrt{1 - \rho_1^2} W_3$$
$$Z_2 = \rho_2 W_2 + \sqrt{1 - \rho_2^2} W_4$$

where W_1, W_2, W_3, W_4 are independent Brownian motions. Then the volatility matrix from Theorem 3.1 is given by

$$\sigma(x_k, k) = \begin{pmatrix} \sqrt{V_k^1} & \sqrt{V_k^2} & 0 & 0 \\ \sigma_1 \sqrt{V_k^1} \rho_1 & 0 & \sigma_1 \sqrt{V_k^1 (1 - \rho_1^2} & 0 \\ 0 & \sigma_2 \sqrt{V_k^2} \rho_2 & 0 & \sigma_2 \sqrt{V_k^2 (1 - \rho_2^2)} \end{pmatrix}$$

so that

$$\sigma\sigma^T = \begin{pmatrix} V_k^1 + V_k^2 & \sigma_1 V_k^1 \rho_1 & \sigma_2 V_k^2 \rho_2 \\ \sigma_1 V_k^1 \rho_1 & \sigma_1^2 V_k^1 & 0 \\ \sigma_2 V_k^2 \rho_2 & 0 & \sigma_2^2 V_k^2 \end{pmatrix}$$

and the drift is given by

$$\mu = \begin{pmatrix} r - q - \frac{1}{2}(V_k^1 + V_k^2) \\ \kappa_1(\theta_1 - V_k^1) \\ \kappa_2(\theta_2 - V_k^2) \end{pmatrix}$$

Then the generator $\mathcal{A}$ as given in Equation 3.2.3 becomes

$$\begin{aligned} \mathcal{A} = &\left(r - q - \frac{1}{2}(V_k^1 + V_k^2)\right) \frac{\partial f}{\partial x_k} + \kappa_1(\theta_1 - V_k^1) \frac{\partial f}{\partial V_k^1} + \kappa_2(\theta_2 - V_k^2) \frac{\partial f}{\partial V_k^2} + \\ &\frac{1}{2}(V_k^1 + V_k^2) \frac{\partial^2 f}{\partial x_k^2} + \rho_1 \sigma_1 V_k^1 \frac{\partial^2 f}{\partial x_k \partial V_k^1} + \rho_2 \sigma_2 V_k^2 \frac{\partial^2 f}{\partial x_k \partial V_k^2} + \frac{1}{2} \sigma_1^2 V_k^1 \frac{\partial^2 f}{\partial {V_k^1}^2} + \\ &\frac{1}{2} \sigma_2^2 V_k^2 \frac{\partial^2 f}{\partial {V_k^2}^2}. \end{aligned} \tag{3.2.6}$$

Substituting $\mathcal{A}$ into Equation 3.2.5 gives us the double Heston model PDE.

Since the Double Heston model belongs to the class of affine models (Christoffersen *et al.* (2009)), meaning that f has a closed-form solution with an exponential affine relationship to the state variables. This is given by the following form

$$\begin{aligned} f(\phi_0, \phi_1, \phi_2; x_k, V_k^1, V_k^2) &= \mathbb{E}\left[\exp(i\phi_0 x_T + i\phi_1 V_T^1 + i\phi_2 V_T^2)\right] \\ &= \exp(A(\tau) + B_0(\tau)x_k + B_1(\tau)V_k^1 + B_2(\tau)V_k^2) \end{aligned} \tag{3.2.7}$$

where $\tau = T - k$.

The coefficients A, B_0, B_1, B_2 can be obtained as follows. We first substitute Equation 3.2.7 for f in Equation 3.2.5 to obtain

$$\begin{aligned} f\Bigg[\left(\frac{\partial A}{\partial k} + \frac{\partial B_0 x_k}{\partial k} + \frac{\partial B_1 V_k^1}{\partial k} + \frac{\partial B_2 V_k^2}{\partial k}\right) + \mu_1 B_0 + \mu_2 B_1 + \mu_3 B_2 + \frac{1}{2}\Bigg((\sigma\sigma^T)_{11}B_0^2 + \\ (\sigma\sigma^T)_{22}B_1^2 + (\sigma\sigma^T)_{33}B_2^2 + (\sigma\sigma^T)_{12}B_0B_1 + (\sigma\sigma^T)_{13}B_0B_2\Bigg)\Bigg] = 0 \end{aligned} \tag{3.2.8}$$

Note that μ and $\sigma\sigma^T$ are affine, such that

$$\begin{aligned} \mu(x_k) &= K_0 + K_1 x_1 + K_2 V_k^1 + K_3 V_k^2 \\ \sigma(x_k)\sigma(x_k)^T &= H_0 + H_1 x_k + H_2 V_k^1 + H_3 V_k^2 \end{aligned}$$

where

$$K_0 = \begin{pmatrix} r - q \\ \kappa_1\theta_1 \\ \kappa_2\theta_2 \end{pmatrix}, \quad K_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad K_2 = \begin{pmatrix} -\frac{1}{2} \\ -\kappa_1 \\ 0 \end{pmatrix}, \quad K_3 = \begin{pmatrix} \frac{-1}{2} \\ 0 \\ -\kappa_2 \end{pmatrix}$$

and

$$H_0 = H_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad H_2 = \begin{pmatrix} 1 & \sigma_1\rho_1 & 0 \\ \sigma_1\rho_1 & \sigma_1^2 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad H_3 = \begin{pmatrix} 1 & 0 & \sigma_2\rho_2 \\ 0 & 0 & 0 \\ \sigma_2\rho_2 & 0 & \sigma_2^2 \end{pmatrix}.$$

Substituting the variables from μ and $\sigma\sigma^T$ in the Equation 3.2.8, we get

$$\begin{aligned} f\Bigg[\frac{\partial A}{\partial k} + \frac{\partial B_0 x_k}{\partial k} + (r-q)B_0 + \kappa_1\theta_1 B_1 + \kappa_2\theta_2 B_2 + \\ \frac{\partial B_1 V_k^1}{\partial k} + V_k^1\left(-\frac{1}{2}B_0 - \kappa_1 B_1 + \frac{1}{2}B_0^2 + \frac{1}{2}\sigma_1^2 B_1^2 + \frac{1}{2}\sigma_1\rho_1 B_0 B_1\right) + \\ \frac{\partial B_2 V_k^2}{\partial k} + V_k^2\left(-\frac{1}{2}B_0 - \kappa_2 B_2 + \frac{1}{2}B_0^2 + \frac{1}{2}\sigma_2^2 B_2^2 + \frac{1}{2}\sigma_2\rho_2 B_0 B_2\right)\Bigg] = 0. \end{aligned}$$

We will drop f because it is always true that $f > 0$. In order for the drift term to equal 0 for all values of x_k, V_k^1 and V_k^2, their coefficient terms and the constants terms must sum to 0. That gives us the following system of ODEs

$$\begin{aligned}
\frac{\partial B_0}{\partial k} &= 0 \\
\frac{\partial A}{\partial k} + (r-q)B_0 + \kappa_1\theta_1 B_1 + \kappa_2\theta_2 B_2 &= 0 \\
\frac{\partial B_1}{\partial k} - \frac{1}{2}B_0 - \kappa_1 B_1 + \frac{1}{2}B_0^2 + \frac{1}{2}\sigma_1^2 B_1^2 + \frac{1}{2}\sigma_1\rho_1 B_0 B_1 &= 0 \\
\frac{\partial B_2}{\partial k} - \frac{1}{2}B_0 - \kappa_2 B_2 + \frac{1}{2}B_0^2 + \frac{1}{2}\sigma_2^2 B_2^2 + \frac{1}{2}\sigma_2\rho_2 B_0 B_2 &= 0
\end{aligned} \tag{3.2.9}$$

These are Riccati equations for the Double Heston model. The solution to these Riccati equations can be found in many textbooks on differential equations. Rouah (2013) provided solutions to the Riccati equations for the Standard Heston model equations on Page 12 and 263. Rouah (2013) also argue that B_1 and B_2 are identical to their counterparts in the Standard Heston model, therefore their solutions are

$$\begin{aligned}
B_0(\tau) &= 0 \\
B_j(\tau,\phi) &= \frac{\kappa_j - \rho_j\sigma_j\phi i + d_j}{\sigma_j^2}\left[\frac{1-e^{d_j\tau}}{1-g_j e^{d_j\tau}}\right] \\
A(\tau,\phi) &= (r-q)\phi i\tau + \sum_{j=1}^{2}\frac{\kappa_j\theta_j}{\sigma_j^2}\left[(\kappa_j - \rho_j\sigma_j\phi i + d_j)\tau - 2\ln\left(\frac{1-g_j e^{d_j\tau}}{1-g_j}\right)\right]
\end{aligned} \tag{3.2.10}$$

where

$$g_j = \frac{\kappa_j - \rho_j\sigma_j\phi i + d_j}{\kappa_j - \rho_j\sigma_j\phi i - d_j}$$

$$d_j = \sqrt{(\kappa_j - \rho_j\sigma_j\phi i)^2 + \sigma_j^2\phi(\phi + i)}$$

for $j = 1, 2$.

With the known coefficients A, B_0, B_1 and B_2, we can now compute the characteristic function f. Christoffersen *et al.* (2009) computed the price of a European call option under the Double Heston model via the Fourier inversion as

$$C(K) = S_k e^{-q\tau}P_1 - Ke^{-r\tau}P_2$$

where K is the strike price, and

$$P_1 = \frac{1}{2} + \frac{1}{\pi}\int_0^\infty \mathrm{Re}\left[\frac{e^{-i\phi\ln K} f(\phi - i; x_k, V_k^1, V_k^2)}{i\phi S_t e^{-\tau}}\right] d\phi$$

$$P_2 = \frac{1}{2} + \frac{1}{\pi}\int_0^\infty \text{Re}\left[\frac{e^{-i\phi \ln K} f(\phi; x_k, V_k^1, V_k^2)}{i\phi}\right] d\phi.$$

The European put options can be obtained via put-call parity.

3.3 State-Space Representations

In order to implement or find the volatility smile of the above mentioned stochastic volatility models, we first have to estimate the unobservable volatilities $V_k, \{V_k^i\}_{i,k}$ at each timestep k as well as the unknown parameters $\{\theta, \sigma, \kappa, \rho, \theta_i, \sigma_i, \kappa_i, \rho_i\}_{i=1,2}$. This is very difficult to do and several different estimation methods are available in the literature. The main problem is that the volatilities are unobservable which makes it hard to evaluate the likelihood function for the stochastic volatility models. Andersen *et al.* (2002), Chernov and Ghysels (2000) use an Efficient Method of Moments, Pan (2002) uses the Generalized Method of Moments to estimate the volatilities and parameters for the stochastic volatility models. Bakshi *et al.* (1997), Christoffersen and Jacobs (2004), Christoffersen *et al.* (2009) use loss functions, and Javaheri *et al.* (2003), Li (2013) use a filtering approach to estimate the volatilities and parameters for the Standard Heston model. Christoffersen *et al.* (2009) in their paper use the loss function to estimate the parameters for the Double Heston model.

In this study, we use filtering combined with the maximum likelihood estimation to estimate the volatilities and parameters for the Standard Heston and Double Heston stochastic volatility models. The basic idea of filtering is to estimate the unobservable volatilities from noise contaminated observations (which can be quoted option prices, implied volatilities or quoted stock prices). We then use the estimated volatilities in the maximum likelihood estimation in order to evaluate the likelihood function. Using optimization methods (such as the Nelder-Mead method, the SQP method or Simulated Annealing method), we estimate the models' parameters that minimize the value of the likelihood function so that the model prices are as close to their market counterparts.

We take the state variables for the Heston model to be V_k, and V_k^1 and V_k^2 for the Double Heston model. The variance processes are taken as the transition equations and the quoted stock returns or option prices are treated as the model observations. Therefore, for us to estimate the unobservable factors V_k, V_k^1, V_k^2 together with the model's parameters, we simply work with the relationship between the stock returns or option prices and the underlying state variables. This is the relationship between the evolution of the measurement equations and the state transition equations. A system of the measurement and transition equations is called the state-space representation of the model. However, the state noise and the measurement noise of the discussed stochas-

tic volatility models are correlated. Since for the filters the process noise and measurement noise must be uncorrelated, we will use Cholesky decomposition to decorrelate the sources of randomness.

Next we discuss the reformulation of our models in state-space representation which involves the specification of the measurement equations and state transition equations. This is a crucial step for us to be able to do the model's estimation using filtering. First we present the state-space form for the Heston model, and lastly for the Double Heston model.

Under the Heston (1993) model, if we take the spot price S_k as the observation and the variance V_k as the state, then the measurement equation is represented by the stock price equation and the state transition equation by the variance process. Recall that the Heston model is represented by a system of equations

$$\begin{aligned} S_k &= \ln S_{k-1} + \left(r - q - \frac{1}{2}V_{k-1}\right)\Delta k + \sqrt{V_{k-1}}\sqrt{\Delta k}W_{k-1} \\ V_k &= V_{k-1} + \kappa\left(\theta - V_{k-1}\right)\Delta k + \sigma\sqrt{V_{k-1}}\sqrt{\Delta k}Z_{k-1}. \end{aligned}$$

where W_k and Z_k are correlated. To eliminate the correlation between these equations, Javaheri (2011) (Page 87-88) shows that the best way to do this is by subtracting from the variance process $f(x_{k-1}, w_k)$ a multiple of the quantity $h(x_k, v_k) - y_k$, which is equal to zero. Writing

$$\begin{aligned} V_k =& V_{k-1} + (\kappa\theta - \kappa V_{k-1})\Delta k + \sigma\sqrt{V_{k-1}}\sqrt{\Delta k}Z_{k-1} \\ & - \rho\sigma\left[\ln S_{k-1} + \left(r - q - \frac{1}{2}V_{k-1}\right)\Delta k + \sqrt{V_{k-1}}\sqrt{\Delta k}B_{k-1} - \ln S_k\right] \end{aligned}$$

which gives

$$\begin{aligned} V_k =& V_{k-1} + \left[(\kappa\theta - \rho\sigma(r-q)) - \left(\kappa - \frac{1}{2}\rho\sigma\right)V_{k-1}\right)\Delta k + \rho\sigma\ln\left(\frac{S_k}{S_{k-1}}\right) + \\ & \sigma\sqrt{1-\rho^2}\sqrt{V_{k-1}}\sqrt{\Delta k}B_{k-1}. \end{aligned} \tag{3.3.1}$$

where

$$B_k = \frac{1}{\sqrt{1-\rho^2}}(Z_k - \rho W_k)$$

and the measurement equation is

$$y_k = \ln S_k = \ln S_{k-1} + \left(r - q - \frac{1}{2}V_k\right)\Delta k + \sqrt{V_k}\sqrt{\Delta k}W_k \tag{3.3.2}$$

Equation 3.3.1 represents the state transition equation and clearly, B_k and W_k are uncorrelated.

Li (2013) suggested that if we take the spot prices S_k and option prices $C(S_k, K)$ as the observations and the variance V_k as the state, then the measurement equations are represented by

$$\ln S_k = \ln S_{k-1} + \left(r - q - \frac{\rho}{\sigma}\kappa\theta\right)\Delta k + \frac{\rho}{\sigma}V_k + \left[\frac{\rho}{\sigma}(\kappa\Delta k - 1) - \frac{1}{2}\Delta k\right]V_{k-1} + \sqrt{1-\rho^2}\sqrt{\Delta k}\sqrt{V_{k-1}}W_k. \tag{3.3.3}$$

$$y_k^0 = g(S_k, V_k, \Theta) + \epsilon_t^0 \tag{3.3.4}$$

where y_k^0 is the observable option prices, with identical independent distributed measurement errors $\epsilon_k^0 \to N(0, \sigma_0^2)$, independent of W_k and Z_k, and $g(\cdot)$ is the theoretical option price computed from the Heston model.

The state transition equations are given by the variance processes

$$\begin{pmatrix} V_k \\ V_{k-\Delta k} \end{pmatrix} = \begin{pmatrix} \kappa\theta\Delta k \\ 0 \end{pmatrix} + \begin{pmatrix} 1-\kappa\Delta k & 0 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} V_{k-\Delta k} \\ V_{k-2\Delta k} \end{pmatrix} + \begin{pmatrix} \sigma\sqrt{\Delta k V_{k-\Delta k}} \\ 0 \end{pmatrix} Z_k$$

See Li (2013) for the derivation of Equations 3.3.3.

Under the Double Heston model, recall that the system equations are

$$\begin{aligned} \ln S_k &= \ln S_{k-1} + \left((r-q) - \frac{1}{2}(V_k^1 + V_k^2)\right)dk + \sqrt{V_k^1}dW_k^1 + \sqrt{V_k^2}dW_k^2, \\ V_k^1 &= V_{k-1}^1 + \kappa_1(\theta_1 - V_{k-1}^1) \triangle k + \sigma_1\sqrt{V_{k-1}^1}\sqrt{\triangle k}Z_k^1, \\ V_k^2 &= V_{k-1}^2 + \kappa_2(\theta_2 - V_{k-1}^2) \triangle k + \sigma_2\sqrt{V_{k-1}^2}\sqrt{\triangle k}Z_k^2. \end{aligned} \tag{3.3.5}$$

Suppose we take the spot price S_k as the observation and the variance processes V_k^1, V_k^2 as the states, then the measurement equation is represented by the stock price $\ln S_k$ in Equation 3.3.5 and the transition equations by the variance processes V_k^1, V_k^2 in Equation 3.3.5. The problem we face when using these equations, the process noise and the measurement noise are correlated, $d[W^1, Z^1]_k = \rho_1 dk$ and $d[W^2, Z^2]_k = \rho_2 dk$. However, for the filtering the process and the measurement noises must be uncorrelated.

Next we derive the measurement equations and the state transition equations for the Double Heston model to fit into the filtering such that the process noise and the measurement noise are uncorrelated.

By Itō's Lemma, we let $x_k = \ln(\frac{S_k}{S_{k-1}})$. This implies that

$$dx_k = \left(r - q - \frac{1}{2}(V_k^1 + V_k^2)\right)dk + \sqrt{V_k^1}dW_k^1 + \sqrt{V_k^2}dW_k^2 \tag{3.3.6}$$

By using the Cholesky decomposition, we set

$$dW_k^1 = \rho_1 dZ_k^1 + \sqrt{1-\rho_1^2}d\tilde{Z}_k^1$$

$$dW_k^2 = \rho_2 dZ_k^2 + \sqrt{1-\rho_2^2}d\tilde{Z}_k^2$$

where $d[Z^1, \tilde{Z}^1] = d[Z^2, \tilde{Z}^2] = 0$

Substituting dW_k^1, dW_k^2 in Equation 3.3.6, we get

$$\begin{aligned} dx_k &= \left(r - q - \frac{1}{2}(V_k^1 + V_k^2)\right)dt + \sqrt{V_k^1}\left(\rho_1 dZ_k^1 + \sqrt{1-\rho_1^2}d\tilde{Z}_k^1\right) + \\ &\quad \sqrt{V_k^2}\left(\rho_2 dZ_k^2 + \sqrt{1-\rho_2^2}d\tilde{Z}_k^2\right) \\ &= \left(r - q - \frac{1}{2}(V_k^1 + V_k^2)\right)dk + \rho_1\sqrt{V_k^1}dZ_k^1 + \sqrt{1-\rho_1^2}\sqrt{V_k^1}d\tilde{Z}_k^1 + \\ &\quad \rho_2\sqrt{V_k^2}dZ_k^2 + \sqrt{1-\rho_2^2}\sqrt{V_k^2}d\tilde{Z}_k^2. \end{aligned} \tag{3.3.7}$$

From Equation 3.2.1, we know that

$$\sqrt{V_k^1}dZ_k^1 = \frac{1}{\sigma_1}\left(dV_k^1 - \kappa_1(\theta_1 - V_k^1)dk\right) \tag{3.3.8}$$

$$\sqrt{V_k^2}dZ_k^2 = \frac{1}{\sigma_2}\left(dV_k^2 - \kappa_2(\theta_2 - V_k^2)dk\right) \tag{3.3.9}$$

By substituting Equation 3.3.8 and 3.3.9 into 3.3.7, we get

$$\begin{aligned} dx_k &= \left(r - q - \frac{1}{2}(V_k^1 + V_k^2)\right)dk + \frac{\rho_1}{\sigma_1}\left(dV_k^1 - \kappa_1(\theta_1 - V_k^1)dk\right) + \\ &\quad \sqrt{1-\rho_1^2}\sqrt{V_k^1}d\tilde{Z}_k^1 + \frac{\rho_2}{\sigma_2}\left(dV_k^2 - \kappa_2(\theta_2 - V_k^2)dk\right) + \sqrt{1-\rho_2^2}\sqrt{V_k^2}d\tilde{Z}_k^2. \end{aligned}$$

Rearranging this and substituting $x_k = \ln(\frac{S_k}{S_{k-1}})$, we get

$$\begin{aligned} \ln S_k &= \ln S_{k-1} + \left(r - q - \frac{\rho_1}{\sigma_1}\kappa_1\theta_1 - \frac{\rho_2}{\sigma_2}\kappa_2\theta_2\right)\triangle k + \frac{\rho_1}{\sigma_1}V_k^1 + \frac{\rho_2}{\sigma_2}V_k^2 + \\ &\quad \left(\frac{\rho_1}{\sigma_1}(\kappa_1 \triangle k - 1) - \frac{1}{2}\triangle k\right)V_{k-1}^1 + \left(\frac{\rho_2}{\sigma_2}(\kappa_2 \triangle k - 1) - \frac{1}{2}\triangle k\right)V_{k-1}^2 + \\ &\quad \sqrt{1-\rho_1^2}\sqrt{V_{k-1}^1}\sqrt{\triangle k}\tilde{Z}_k^1 + \sqrt{1-\rho_2^2}\sqrt{V_{k-1}^2}\sqrt{\triangle k}\tilde{Z}_k^2. \end{aligned} \tag{3.3.10}$$

which is the measurement equation.

The state transition equations are:

$$\begin{pmatrix} V_k^1 \\ V_k^2 \end{pmatrix} = \begin{pmatrix} V_{k-1}^1 + \kappa_1(\theta_1 - V_{k-1}^1) \triangle k + \sigma_1 \sqrt{V_{k-1}^1} \sqrt{\triangle k} Z_k^1 \\ V_{k-1}^2 + \kappa_2(\theta_2 - V_{k-1}^2) \triangle k + \sigma_2 \sqrt{V_{k-1}^2} \sqrt{\triangle k} Z_k^2 \end{pmatrix} \tag{3.3.11}$$

Clearly, the measurement noise $\tilde{Z}_k^1$ and $\tilde{Z}_k^2$ from in Equation 3.3.10 are uncorrelated to the states noise Z_k^1, Z_k^2 in Equation 3.3.11.

Having reformulated the measurement equations and the state transition equations for a model, its very important to know if the measurement equations contains enough information to allow the estimation of the states. We need to check if it is possible to estimate the states from the given measurement equation(s). If the measurement equation $y = 0$, for instance, then it is not possible to find the state variables from this measurement equation, because it contains not enough information about the states. A system that allows states estimation from the measurement equation(s) is referred to as observable. We therefore need to check if the measurement equation given in Equation 3.3.10 and the states in Equation 3.3.11 form an observable system. Below we provide a mathematical definition of the *observability* of a system.

A nonlinear system with a state vector x_k of dimension n is observable if

$$\mathbf{O} = \begin{pmatrix} H \\ HA \\ HA^2 \\ \vdots \\ HA^{n-1} \end{pmatrix}$$

has a full rank of n. H and A are the Jacobian matrices of the measurement and the states as defined in Chapter 2, under the extended Kalman filter. For more details on Observability, see Reif *et al.* (1999), Sira-Ramirez (1988) and Hermann and Krener (1977).

Now we have a look at the observability of the measurement and transition equations in Equations 3.3.10 and 3.3.11 respectively.

Since

$$H = \left(\tfrac{\rho_1}{\sigma_1}(\kappa_1 \triangle k - 1) - \tfrac{1}{2} \triangle k \qquad \tfrac{\rho_2}{\sigma_2}(\kappa_2 \triangle k - 1) - \tfrac{1}{2} \triangle k\right),$$

then the observation matrix is

$$\mathrm{O} = \begin{pmatrix} \tfrac{\rho_1}{\sigma_1}(\kappa_1 \triangle k - 1) - \tfrac{1}{2} \triangle k & \tfrac{\rho_2}{\sigma_2}(\kappa_2 \triangle k - 1) - \tfrac{1}{2} \triangle k \\ \left(\tfrac{\rho_1}{\sigma_1}(\kappa_1 \triangle k - 1) - \tfrac{1}{2} \triangle k\right)(1 - \kappa_1 \triangle k) & 0 \end{pmatrix}.$$

For $\triangle k = 0$, then

$$\mathrm{O} = \begin{pmatrix} \frac{\rho_1}{\sigma_1} & \frac{\rho_2}{\sigma_2} \\ \frac{\rho_1}{\sigma_1} & 0 \end{pmatrix},$$

$\det(O) = -\frac{\rho_1\rho_2}{\sigma_1\sigma_2} < 0$. So for small $\triangle k$, $\det(O) \neq 0$ and hence O is non-singular for small $\triangle k$ and since it is a 2×2, then its rank must be 2. Therefore our system is observable.

Now suppose we take the spot price S_k and the option prices $C(S_k, K)$ as the observations and the variance processes V_k^1, V_k^2 as the states, then the measurement equations can be derived in the same way as was done for the Standard Heston model in Li (2013). We simply replace the Heston model theoretical option price $g(\cdot)$ with the one for the Double Heston model. These measurement and the state transition equations are then used in the extended, unscented Kalman filter and the particle filter to estimate the model's volatilities and the parameters.

Chapter 4

Empirical Analysis

In Chapter 2, we discussed filtering methods. We also discussed two stochastic volatility models and their characteristic functions in Chapter 3. The idea is to implement these stochastic volatility models which have a volatility driven by a time-varying variance process. Unlike the Standard Heston model, the Double Heston model is driven by two variance processes. These variance processes consist of unknown parameters. Therefore, implementing stochastic volatility models is difficult since the volatilities are unobservable and parameters are hidden.

In this chapter we use the three non-linear filtering methods on options data. With these methods we extract the model's implied volatilities and compare their performance. We also combine the filters with the maximum likelihood estimation method to estimate the hidden parameters.

In Section 4.1 we present our data set, Section 4.2 discusses about the Jacobian matrices in the extended Kalman filter. Section 4.3 details the estimated parameters under the Standard Heston model as well as the comparisons of the term structures of the extracted implied volatilities and the market implied volatilities. Section 4.4 presents the estimated parameters under the Double Heston model as well as the term structures of the extracted implied volatilities and the market implied volatilities. All the experiments are done in Matlab.

4.1 Data

In this study, the data used is put option prices on the Dow Jones Industrial Average ETF (DJIA), recorded on May 10, 2012. There are four maturities 37, 72, 135, and 226 days. The closing price is \$129.14 and the strikes prices ranges from $K = 124$ to $K = 136$ in increments of \$1. The options dataset is quoted as implied volatility, as presented in Table 4.1.

Table 4.1: DJIA Implied volatilities on 10 May 2012

Strike	Maturity(Days)			
K	37	72	135	226
124	0.1962	0.1947	0.2019	0.2115
125	0.1910	0.1905	0.1980	0.2082
126	0.1860	0.1861	0.1943	0.2057
127	0.1810	0.1812	0.1907	0.2021
128	0.1761	0.1774	0.1871	0.200
129	0.1718	0.1743	0.1842	0.1974
130	0.1671	0.1706	0.1813	0.1950
131	0.1644	0.1671	0.1783	0.1927
132	0.1645	0.1641	0.1760	0.1899
133	0.1661	0.1625	0.1743	0.1884
134	0.1701	0.1602	0.1726	0.1862
135	0.1755	0.1610	0.1716	0.1846
136	0.1796	0.1657	0.1724	0.1842

We obtain this dataset from the Fabrice Douglas Rouah website Rouah (2013 (accessed September 3, 2015) in his MATLAB codes titled "the Heston model and its extensions in Matlab and C#", chapter 9, script name: Mikhailov Nogel estimation DJIA. We use this data in our study for empirical analysis only, different quoted data can be used for implementations.

Since the options dataset is quoted as implied volatilities, to obtain the market put prices, we use the Black-Scholes formula,

$$C = Se^{-qT}N(d_1) - Ke^{-rT}N(d_2),$$
$$P = Ke^{-rT}N(-d_2) - Se^{-qT}N(-d_1),$$

where C and P are the call and put prices, respectively. The stock closing price is denoted by S, q is the expected dividend rate, r is a risk-free interest rate, T is the maturity, and K is the strike price. We also denoted the cumulative probability distribution function for standard normal distribution as $N(\cdot)$, with the d_1 and d_2 defined as follows

$$d_1 = \frac{\ln(S/K) + (r - q + V^2/2)T}{V\sqrt{T}}$$

and

$$d_2 = d_1 - V\sqrt{T},$$

where V is the quoted implied volatility.

4.2 Fitting filters

In Chapter 3 we have derived the state-space representations of the Standard Heston (1993) and Double Heston models. The state transition and measurement equations for these models are uncorrelated. All the filters were initialized at the unconditional means and covariance of the states. For the Standard Heston (1993) model, the mean was initialized by $\hat{x}_0 = V_0$ and the initial covariance $P_0^- = \text{diag}(\sigma^2 \Delta k V_0, 0)$, where V_0 is the initial level of the variance. Recall that $\hat{x}_k$ are the unobservable volatilities.

For the Double Heston model, the initial mean is

$$\hat{x}_0 = \left[V_0^1, V_0^2\right]^T$$

and the covariance $P_0^- = \text{diag}(\sigma_1^2 \Delta k V_0^1, \sigma_2^2 \Delta k V_0^2)$, where V_0^1 and V_0^2 are the initial levels of the variance processes.

The Standard Heston model parameters to be estimated are $\kappa, \theta, \sigma, V_0, \rho$. The parameters for the Double Heston model are $\kappa_1, \theta_1, \sigma_1, V_0^1, \rho_1, \kappa_2, \theta_2, \sigma_2, V_0^2, \rho_2$. In the estimation, we keep the parameters fixed at all the maturities. We first initialize the parameters and use the filters to estimate the volatilities and then use the MLE for valuation of the likelihood function. Optimization routines (see Section 2.4) are then used to estimate the best parameters that maximize the likelihood function.

The volatilities $\hat{x}_k$ are estimated using the filtering techniques discussed in Chapter 2. The state transition and measurement equations are as discussed in Section 3.3.1. In this study we use option prices only as the observations, with the put option price given by

$$\begin{aligned} y_k^0 &= g(S, V_k, \Theta) + \epsilon_t^0 \\ &= C(K) + Ke^{-r\tau} - S_k e^{-q\tau} + \epsilon_t^0, \end{aligned} \tag{4.2.1}$$

where y_k^0 represents the put option price under the Standard Heston model, $C(K)$ is the call price as defined in Equation 3.1.3, and Θ is the set of model parameters.

The Jacobian matrices in the extended Kalman filter under the Heston model are given by

$$\begin{aligned} A_k &= 1 - \kappa \Delta k, \\ W_k &= \sigma \sqrt{V_{k-1}} \sqrt{\Delta k} \end{aligned}$$

To compute the gradient matrix H_k of y_k^0 with respect to the implied volatility is not straight forward. We follow the approach of Zhu (2009) (Page 82-83).

Rouah (2013) (Page 329) also used the same approach. Since the stock returns volatility is driven by a variance process which consists of parameters, Zhu (2009) recommends the derivatives of the call and put price with respect to the implied volatility to be based on two parameters, $\nu_1 = V_0$ the initial level of the variance, and $\nu_2 = \theta$ the long term level of the variance. Therefore, the Jacobian matrices of the put option with respect to the volatility

$$\begin{aligned} H_1 &= \frac{\partial y_k^0}{\partial \nu_1} = \frac{\partial y_k^0}{\partial V_0} 2\sqrt{V_0} \\ H_2 &= \frac{\partial y_k^0}{\partial \nu_2} = \frac{\partial y_k^0}{\partial \theta} 2\sqrt{\theta} \end{aligned} \tag{4.2.2}$$

Substitute the y_k^0 from Equation 4.2.1 into Equation 4.2.2, yields

$$H_1 = Se^{-q\tau}\frac{\partial P_1}{\partial V_0}2\sqrt{V_0} - Ke^{-r\tau}\frac{\partial P_2}{\partial V_0}2\sqrt{V_0}$$

where

$$\frac{\partial P_j}{\partial V_0} = \frac{1}{\pi}\int_0^\infty \text{Re}\left[\frac{e^{-i\phi \ln K} f_j(\phi; S_k, V_k) B_j(\tau, \phi)}{i\phi}\right] d\phi,$$

for $j = 1, 2$ and $B_j(\tau, \phi)$ as defined in Section 3.1. The derivative of the second Jacobian matrix,

$$H_2 = Se^{-q\tau}\frac{\partial P_1}{\partial \theta}2\sqrt{\theta} - Ke^{-r\tau}\frac{\partial P_2}{\partial \theta}2\sqrt{\theta}$$

where

$$\frac{\partial P_j}{\partial \theta} = \frac{1}{\pi}\int_0^\infty \text{Re}\left[\frac{e^{-i\phi \ln K} f_j(\phi; S_k, V_k) A_j(\tau, \phi)/\partial\theta}{i\phi}\right] d\phi$$

and

$$\frac{\partial A_j(\tau, \phi)}{\partial \theta} = \frac{\kappa}{\sigma^2}\left[(b_j - \rho\sigma\phi i + d_j)\tau - 2\ln\left(\frac{1 - g_j e^{d_j\tau}}{1 - g_j}\right)\right].$$

Under the Double Heston model, the put option price is given by

$$\begin{aligned} y_k^0 &= g(S, V_k^1, V_k^2, \Theta) + \epsilon_t^0 \\ &= C(K) + Ke^{-r\tau} - S_k e^{-q\tau} + \epsilon_t^0. \end{aligned} \tag{4.2.3}$$

The Jacobian matrices in the extended Kalman filter under the Double Heston model are given by

$$\begin{aligned} A_1 &= 1 - \kappa_1 \Delta k, \quad A_2 = 1 - \kappa_2 \Delta k \\ W_1 &= \sigma_1\sqrt{V_k^1}\sqrt{\Delta k}, \quad W_2 = \sigma_2\sqrt{V_k^2}\sqrt{\Delta k} \end{aligned}$$

Since the Double Heston model has two initial variance parameters V_0^1 and V_0^2, then the derivatives of the put price with respect to the implied volatility is based on $\nu_1 = \sqrt{V_0^1}$ and $\nu_2 = \sqrt{V_0^2}$. These Jacobian matrices are given as

$$H_j = Se^{-q\tau}\frac{\partial P_1}{\partial V_0^j}2\sqrt{V_0^j} - Ke^{-r\tau}\frac{\partial P_2}{\partial V_0^j}2\sqrt{V_0^j}$$

where

$$\frac{\partial P_1}{\partial V_0^j} = \frac{1}{\pi}\int_0^\infty \text{Re}\left[\frac{e^{-i\phi\ln K}}{i\phi S_k e^{(r-q)\tau}} f(\phi - i; S_k, V_k^1, V_k^2)B_j(\tau, \phi - i)\right] d\phi,$$
$$\frac{\partial P_2}{\partial V_0^j} = \frac{1}{\pi}\int_0^\infty \text{Re}\left[\frac{e^{-i\phi\ln K}}{i\phi} f(\phi; S_k, V_k^1, V_k^2)B_j(\tau, \phi)\right] d\phi,$$

for $j = 1, 2$ and $B_j(\tau, \phi)$ as defined in Equation 3.2.10.

For the computation of H_k's we use the central-difference method. The calculation of the put price requires the evaluation of the integral of the characteristic functions. We use a numerical approximation method, the Gauss-Laguerre Quadrature. This method is well-suited in our integral, because it does not require lower and upper limits of the integration. In our case the integration domain is $(0, \infty)$ which can be challenging in choosing the lower and upper limits.

4.3 Parameters in the Heston model

The estimated parameters $\kappa, \theta, \sigma, V_0, \rho$ of the Standard Heston model from the extended Kalman filter (EKF), the unscented Kalman filter (UKF) and the particle filter (PF) are found in Table 4.2. We estimated the parameters using options on strike prices $K = 124 : 134$ only, leaving out options on strike prices 135 and 136.

Table 4.2: Estimated parameters for the Heston model

	κ	θ	σ	V_0	ρ
EKF	0.2898	0.1258	1.4002	0.0356	-0.426
UKF	0.7782	0.2012	1.6699	0.0369	-0.4107
PF	1.3421	0.1304	1.3568	0.0356	-0.4192

We obtained nearly the same parameters from both optimizations methods, therefore we only present the parameters from the `fminsearch` optimizer. The parameters for the extended Kalman filter and the particle filters are nearly

the same. It is not necessary for different filters to have same parameters. The mean reversion speed κ and the volatility of the variance σ control curvature of the volatility smile. A higher value of κ flatters the volatility smile and a higher value of σ increases the curvature of the smile. The parameters θ and V_0 determine the level of the smile, with a higher value of V_0 leading to a higher volatility σ. The correlation parameter ρ controls the direction of the implied volatility smile. A negative ρ yields a negative slope of the smile and a positive ρ yields a positive slope.

Fig. 4.1 plots the term structure for the market implied volatilities and the extracted implied volatilities under the Standard Heston model.

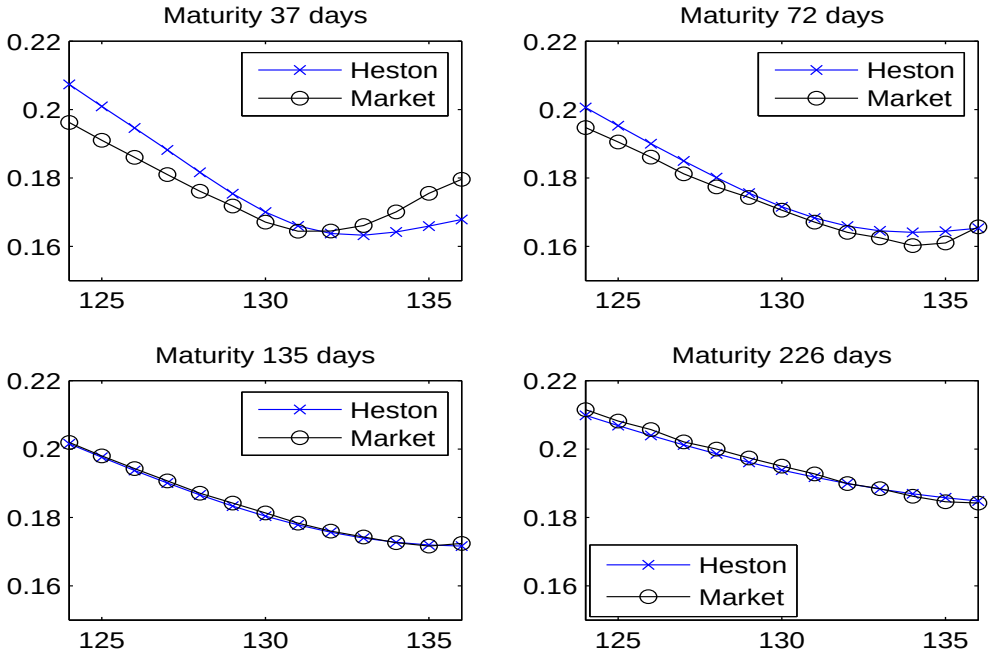

Figure 4.1: Heston implied volatilities from EKF

The implied volatilities in Fig. 4.1 from the extended Kalman filter do not fit the actual implied volatilities well especially at short maturities. In the maturities of 37 and 72 days, the Heston model completely failed to capture the market implied volatilities. At long maturities (135 and 226 days) we get a better fit compared to 37 and 72 days. The extended Kalman filter has a lower κ which saddles the implied volatilities smile.

The term structure for the implied volatilities of the Standard Heston model extracted by the unscented Kalman filter are found in Fig. 4.2. As shown in

Fig. 4.2, the unscented Kalman filter provides a better fit for the volatility smile at long maturities, 135 and 226 days. The short maturities still failed to capture the smile. In the unscented Kalman filter the computation was done setting $\alpha = 10^{-3}$, $\kappa = 0$ and $\beta = 2$.

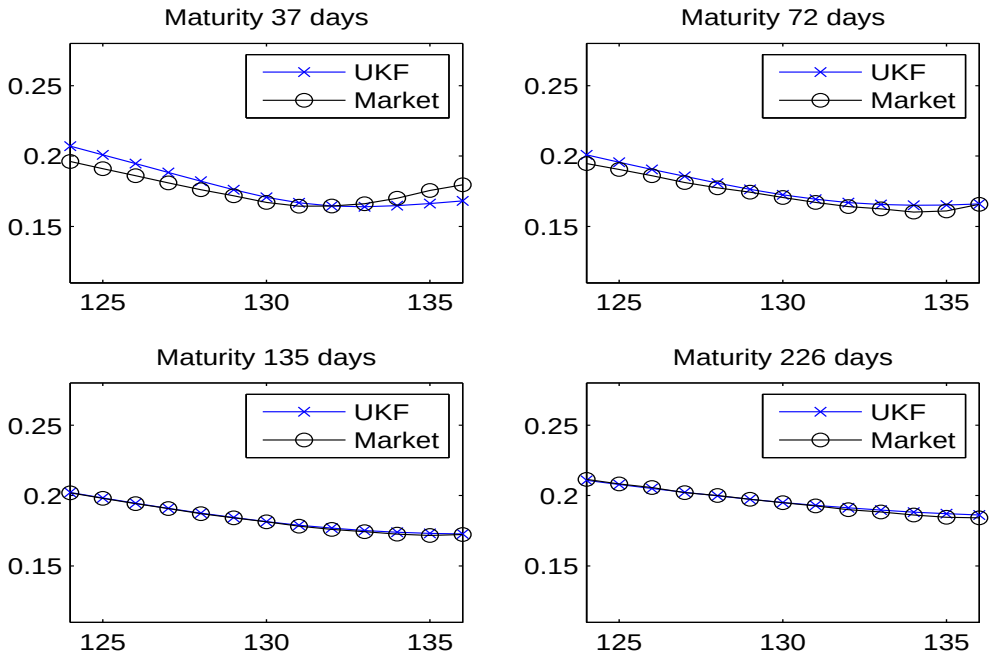

Figure 4.2: Heston implied volatilities from UKF

The term structure for the implied volatilities of Heston model extracted using particle filter are found in Fig. 4.3. Based on the results in Fig. 4.2 and 4.3, it can be difficult to notice the difference in the implied volatilities given by the unscented Kalman and particle filters. However possible differences, if any, should become clear once a highly non-linear or a non-Gaussian model is introduced. In the particle filter 500 particles were used and this was more computational demanding and took longer than the unscented Kalman filter.

When we estimated the parameters in Table 4.2 we left out option prices for the strike prices $K = 135$ and 136. In Table 4.3 we present the Heston model put option prices for the strike prices $K = 135$ and 136. These put prices are computed using estimated parameters in Table 4.2. Each strike price has four maturity days, namely 37, 72, 135, and 226. The closing price is $S = 129.14$.

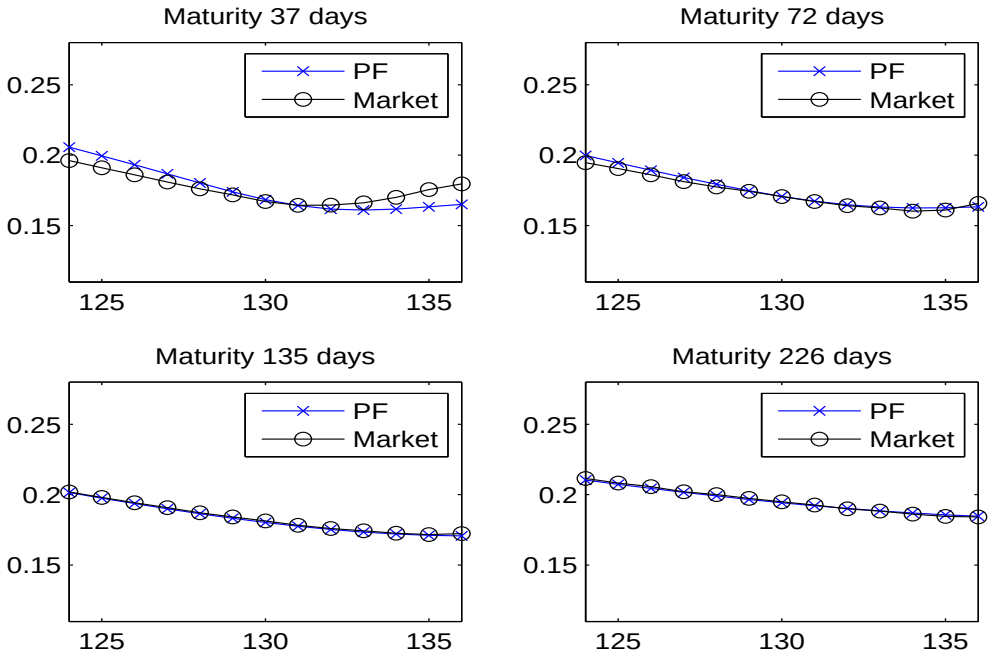

Figure 4.3: Heston implied volatilities from PF

Table 4.3: Put option prices from the Heston model

Filter	Strike (K)	Maturity (Days) 37	72	135	226
Actual Data	135	6.814255	7.504627	9.091425	11.199449
	136	7.674530	8.354330	9.816700	11.849407
EKF	135	7.084522	7.849682	9.021301	10.594243
	136	7.941194	8.681519	9.814157	11.342153
UKF	135	6.668625	7.434077	8.771653	11.384425
	136	7.521106	8.228892	9.488230	11.335073
PF	135	6.636029	7.471980	8.953781	11.027706
	136	7.478933	8.239642	9.635039	11.644477

4.4 Parameters in the Double Heston model

The estimated parameters $\kappa_1, \theta_1, \sigma_1, V_0^1, \rho_1, \kappa_2, \theta_2, \sigma_2, V_0^2, \rho_2$ for the extended Kalman filter (EKF), the unscented Kalman filter (UKF) and the particle filter (PF) under the two-factor model are found in Table 4.4. Similarly to the Standard Heston model, the parameters in the Double Heston model are also estimated using options on strike prices $K = 124 : 134$ only, leaving out options on strike prices 135 and 136.

Table 4.4: Estimated parameters for the Double Heston model

Parameter	EKF	UKF	PF
κ_1	3.0769	3.2381	3.0853
θ_1	0.0312	0.0307	0.0310
σ_1	2.0207	1.9898	2.0023
V_0^1	0.0261	0.02721	0.0260
ρ_1	-0.0654	0.0604	0.0649
κ_2	1.8668	1.8131	1.8628
θ_2	0.0600	0.0583	0.0610
σ_2	0.6982	0.7105	0.7021
V_0^2	0.0093	0.0095	0.0093
ρ_2	-0.9925	-1.0	-0.9836

As shown in Table 4.4 the parameters for all the filters are nearly the same. From Table 4.2 the correlation parameter ρ under the Heston model is always negative. Under the Double Heston model the correlation parameters ρ_1 and ρ_2 can be negative or positive as shown on Table 4.4. The empirical studies also show that the market data yields negative as well as positive correlations. We therefore expect the Double Heston model to perform better than the Standard Heston model. The long-term variance level θ_2 of the second volatility factor controls the level of the implied volatility smile at short maturities and θ_1 controls the level of the smile for longer maturities.

Fig. 4.4 plots the implied volatilities smile of the Double Heston model extracted using the extended Kalman filter and the market implied volatilities smile. The results show a poor fit of the extended Kalman filter on the Double Heston model. This also shows the extended Kalman filter weakness on highly non-linear models. The linearization of the Double Heston model put option equation might have also contributed errors. The volatility smile under the Double Heston model at shorter maturities is better than under the Standard Heston model.

In Fig. 4.5 we present the implied volatilities under the Double Heston model estimated by the unscented Kalman filter. The estimates are better than for the extended Kalman filter. During estimation we set $\alpha = 0.99$, $\kappa = 0$ and $\beta = 2$, which implies that under the Double Heston model the sigma points are widely spread for better estimates. Implied volatilities estimated via the unscented Kalman filter do not fit the market implied volatilities very well especially in the longer maturity of 226 days as shown in Fig. 4.5.

Fig. 4.6 displays the implied volatilities under the Double Heston model estimated by the particle filter. We obtained nearly a perfect fit at all maturi-

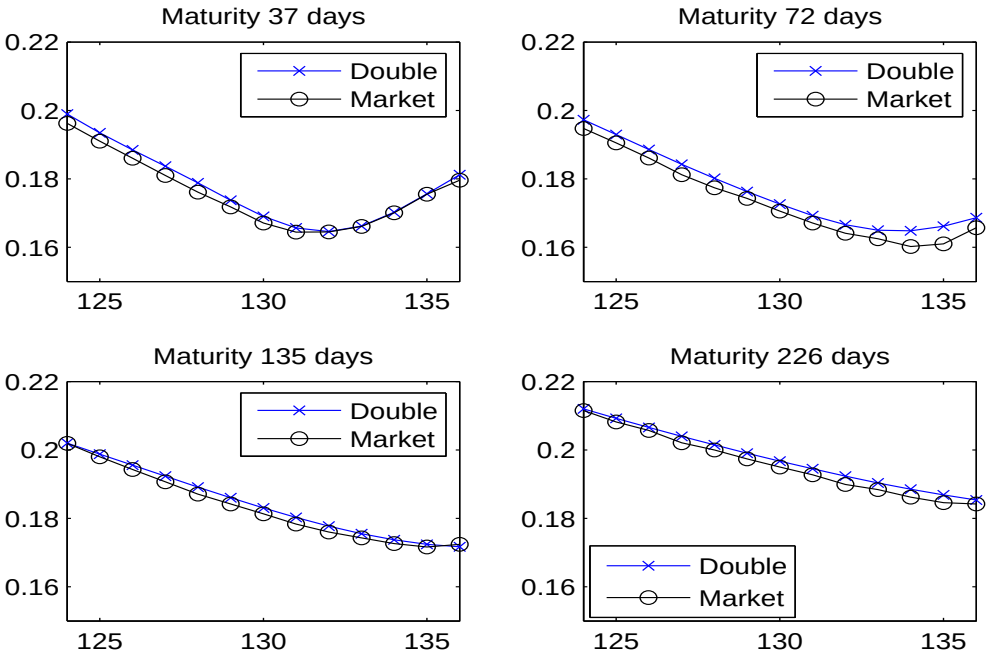

Figure 4.4: Double Heston model Implied volatilities under EKF

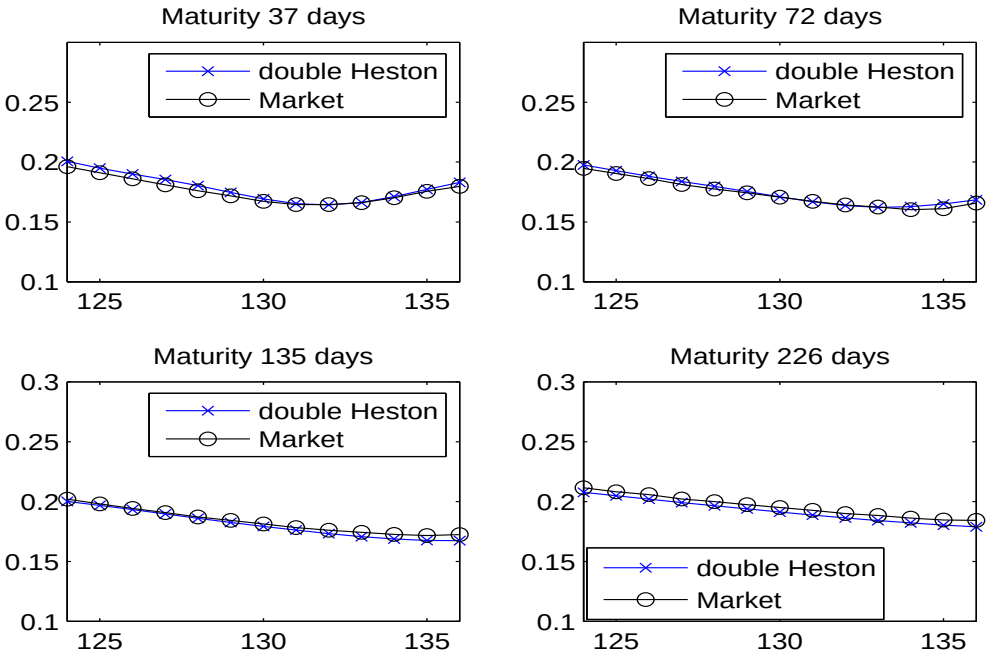

Figure 4.5: The Double Heston model implied volatilities under UKF

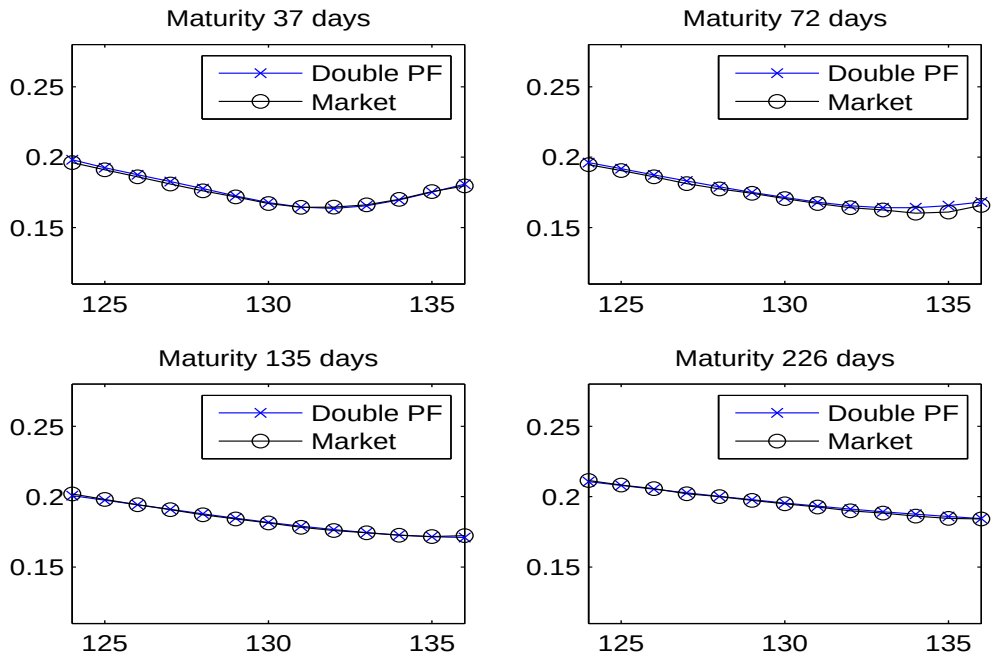

Figure 4.6: The Double Heston model implied volatilities under PF

ties. Furthermore, the Standard Heston model could not capture the smile at shorter maturities. Therefore, the Double Heston model offers more flexibility in modelling the volatility than the Standard Heston model. In the iteration we used 500 particles.

Table 4.5 displays the Double Heston model put option prices for strike prices $K = 135$ and 136. We computed the prices using the estimated parameters in Table 4.4. Compared to the Heston model prices in Table 4.3, the Double Heston model put prices are closer to the actual prices. The unscented Kalman filter and the particle filter also performed better than the extended Kalman filter in both models.

Table 4.5: Put option prices from the Double Heston model

Filter	Strike (K)	Maturity (Days) 37	72	135	226
Actual Data	135	6.814255	7.504627	9.091425	11.199449
	136	7.674530	8.354330	9.816700	11.849407
EKF	135	6.834805	7.580502	8.973786	11.033095
	136	7.713028	8.403014	9.672677	11.642593
UKF	135	6.812425	7.578929	9.059128	11.200290
	136	7.691932	8.389437	9.743906	11.808492
PF	135	6.810771	7.592233	9.091525	11.249225
	136	7.688207	8.399582	9.778391	11.860654

Chapter 5

Conclusion

Many stochastic volatility models have been developed in the last decade, but the standard Heston model has emerged as the most popular stochastic volatility model in option pricing theory. The Double Heston model which was proposed recently has also attracted the attention of traders. In this dissertation we have studied and implemented these two stochastic volatility models. In general, the implementation of the stochastic volatility models is quite challenging, as one is confronted with the difficulty of jointly estimating the models' hidden parameters, as well as the unobservable volatilities. The unobservable volatilities were estimated through the filtering approach using market option prices. Filtering methods are considered to be efficient in distinguishing unobservable state variables from noisy observations, regardless of the observations size. These methods converge faster to the true values compared to other methods, like Monte Carlo, that require large numbers of simulations for better estimation.

The study began with a brief discussion of the filtering methods. For better understanding of the filters we began with the Kalman filtering. This filtering method was not used in the implementation, because our models are non-linear and it is only optimal for linear models. The other filtering methods discussed are the extended Kalman filter, the unscented Kalman filter and the particle filter. Our numerical results for comparison show that the particle filter performs better compared to the other filters. Under the Standard Heston model, the implied volatilities from the particle filter and the unscented Kalman filter are almost the same. The difference between the unscented Kalman filter and the particle filter became clear under the Double Heston model. The extended Kalman filter performed poorly for both models. The results also show that the Double Heston model fits the market implied volatility smile very well at all maturities. The Standard Heston model fails to explain the random movements in the implied volatilities smile at shorter maturities. Therefore, the Double Heston model has introduced more flexibility in modelling the volatility.

The particle filter has been criticised for its degeneracy problem. To avoid degeneracy, one can consider a good choice of the proposal distribution or including the resampling step in the algorithm. In this study we used both the proposal distribution and the resampling step, and this makes particle filtering more efficient. The approximation of the Jacobian matrices in the extended Kalman filter are done using the central-difference method. The maximum likelihood estimation method was used to estimate the model's parameters.

Future work includes the use of filters on estimating implied volatilities as well the parameters under the stochastic volatility models using options on FTSE/JSE Top 40 index (ALSI) quoted on 05 Feb 2013. We already obtained the data from Johannesburg Stock Exchanges (JSE).

Appendix A

Kalman filter

This appendix derives the prediction and update equations in the Kalman filter algorithm. The derivations were created by my project supervisor. We first start with some properties in probability.

A.1 Probability properties

If $Z = X + Y$, all three being vectors of the same length whose elements are random variables with zero means, and X, Y are independent then

$$cov(Z) = cov(X) + cov(Y)$$

where $cov(X) = E\left(XX^T\right)$ and similarly for Y, Z. If X is deterministic then $cov(Z) = cov(Y)$. If X does not have zero mean then

$$cov(X) = E((X - E(X))(X - E(X))^T)$$

If X, Y have mean $\bar{X}, \bar{Y}$ then

$$E((X - \bar{X})(Y - \bar{Y})^T) = E(XY^T) - \bar{X}\bar{Y}^T$$

If X has zero mean and A a deterministic matrix such that AX is defined then

$$cov(AX) = E((AX)(AX)^T) = E(AXX^TA^T) = AE(XX^T)A^T = Acov(X)A^T.$$

If $X \sim N(\mu, \Sigma$ is normal then AX is also normal and $AX \sim N(A\mu, A\Sigma A^T)$.

If X, Y have lengths m, n respectively then $cov(X, Y)$ is an $m \times n$ matrix.

Let $X, Z, (X, Z)^T$ be vectors of random variables, with Z multivariate normal satisfying

$$\begin{pmatrix} X \\ Z \end{pmatrix} = N\left(\begin{pmatrix} \mu_X \\ \mu_Z \end{pmatrix}, \begin{pmatrix} A & B \\ C & D \end{pmatrix}\right)$$

where A, D are square and have the same number of rows as X, Y respectively, and D is invertible.

Then $A = cov(X) = P^{xx}, D = cov(Z) = P^{zz}, B = cov(X, Z) = P^{xz}, C = cov(Z, X) = P^{zx}$.

If X conditioned on $Z = z$ is written as $(X|Z = z)$, then it satisfies

$$(X|Z = z) \sim N(\mu_X + BD^{-1}(z - \mu_Z), A - BD^{-1}C)$$

A.2 Kalman filtering

Consider $n-$vectors $\{x_k\}, \{y_k\}, \{w_k\}, \{v_k\}$, deterministic $n \times n$ matrices M and $\{H_k\}$ and $n \times n$ positive definite matrices Q, R $(k = 0, 1, 2, \dots)$, and deterministic $n-$vectors C_k, D_k.

x_k is the internal state of a system and satisfies

$$x_k = Ax_{k-1} + C_k + w_k$$

x_0 may be unknown.

y_k is an output or observation vector, depend on the state, given by

$$y_k = H_k x_k + D_k + v_k$$

$y_0 = H_0 x_0$. Given y_k we may not be able to determine the internal state x_k.

w_k, v_k are noise processes independent of the state, the observation and each other satisfying

$$w_k \sim N(0, Q), \quad v_k \sim N(0, R)$$

(N is the $n-$dimensional normal distribution).

Let

$$I_k = \{\hat{x}_k, P_k, y_k\}$$

denote information available at time k, $\hat{x}_k = E(x_k|y_k, I_{k-1}$ is an estimate of x_k and P_k is an estimate of the covariance of the error in estimating x_k, $cov(x_k - \hat{x}_k)$. y_k is also available at this time. Note that the quantities in I_k are observations and estimates which depend in general only on the initial state and the observations $y_1, \dots, y_k$. So when taking conditional expectations with respect to I_k it would be equivalent to taking conditional expectations

wrt information $\{y_1, \ldots, y_k\} \cup \{\hat{x}_0, P_0\}$.

Let

$$\begin{aligned}\hat{x}_k^- &= E(x_k|I_{k-1}) \\ &= E(Ax_{k-1} + C_k + w_k|I_{k-1}) \\ &= A\hat{x}_{k-1} + C_k\end{aligned}$$

be the best estimate at time $k-1$ of x_k given all the prior information but not y_k. It is called the *prior estimate* of x_k. Here we have used a notation to be introduced later: $\hat{x}_{k-1} = E(x_{k-1}|I_{k-1})$.

The prior estimate of y_k is

$$\begin{aligned}\hat{y}_k^- &= E(y_k|I_{k-1}) \\ &= E(Hx_k + D_k + v_k|I_{k-1}) \\ &= H\hat{x}_{k-1}^- + D_k\end{aligned}$$

The prior error estimate of x_k is

$$\begin{aligned}e_k^- &= x_k - \hat{x}_k^- \\ &= (Ax_{k-1} + C_k + w_k) - (A\hat{x}_{k-1} + C_k \\ &= A(x_{k-1} - \hat{x}_{k-1}) + w_k\end{aligned}$$

$$cov(e_k^- cov(e_k^-|I_{k-1})) = Acov(x_{k-1} - \hat{x}_{k-1}|I_{k-1})A^T + Q$$

where we used material from Section A.1 to compute $cov(e_k^-)$ above. Since $E(x_k|I_{k-1}) = \hat{x}_k^-$ and $E(x_k|y_k, I_{k-1}) = \hat{x}_k$, this can also be written as

$$cov(x_k|I_{k-1}) = Acov(x_{k-1}|I_{k-1})A^T + Q \triangleq AP_{k-1}A^T + Q$$

where P_k will be defined again (equivalently) below.

The prior covariance matrix of the prior error is (noting that the mean of e_k^- is zero)

$$P_k^- = E(e_k^- e_k^{-T}|I_{k-1})(= P_k^{xx-})$$

The prior error estimate of y_k using $\hat{y}_k^-$ is

$$\begin{aligned}e_k^{y-} &= y_k - \hat{y}_k^- \\ &= (H_k x_k + D_k + v_k) - H_k\hat{x}_k^- \\ &= H_k e_k^- + D_k + v_k\end{aligned}$$

The prior covariance matrix of the measurement forecast error e_k^{y-} is therefore (using the probability definitions in Section A.1)

$$F_k = H_k P_k^- H_k^T + R = cov(y_k)(= P_k^{yy-})$$

$$\begin{aligned} cov(e_k^-, e_k^{y-}) &= cov(e_k^-, H_k e_k^- + D_k + v_k) \\ &= cov(e_k^-, H_k e_k^-) \\ &= E(e_k^- e_k^{-T} H_k^T | I_{k-1}) \\ &= E(e_k^- e_k^{-T} | I_{k-1}) H_k^T \\ &= P_k^- H_k^T (= P_k^{xy-}) \end{aligned}$$

Let

$$\hat{x}_k = E(x_k | y_k, I_{k-1})$$

be the best estimate at time k of x_k given present value of y_k and all prior information. It is called the *posterior estimate* of x_k.

The posterior error estimate of x_k is

$$e_k = x_k - \hat{x}_k$$

The posterior covariance matrix of the posterior error is (noting that the mean of e_k is 0)

$$P_k = E(e_k e_k^T | y_k, I_k)(P_k^{xx})$$

As in the discussion of P_k^- we have $P_k = cov(x_k | y_k, I_{k-1})$.

We also introduce the *Kalman Gain*, K_k:

$$\begin{aligned} \hat{x}_k &= \hat{x}_k^- + K_k(y_k - \hat{y}_k^-) \\ &= (I - K_k H_k)\hat{x}_k^- - K_k D_k + K_k y_k \\ &= L_k \hat{x}_k^- - K_k D_k + K_k y_k \\ &= L_k \hat{x}_k^- - K_k D_k + K_k(H_k x_k + v_k) \end{aligned}$$

where $L_k = I - K_k H_k$.

$$\begin{aligned} e_k &= x_k - \hat{x}_k \\ &= L_k(x_k - \hat{x}_k^-) + K_k D_k - K_k v_k \\ &= L_k e_k^- + K_k D_k - K_k v_k \\ P_k &= E(e_k e_k^T | y_k, I_k) = L_k P_k^- L_k^T + K_k R K_k^T \qquad \text{(Josephson Form for } P_k\text{)} \\ &= P_k^- - K_k(H_k P_k^-) - (P_k^- H^T) K_k^T + K_k(H_k P_k^- H_k^T + R) K_k^T \end{aligned}$$

From the state and measurement equations it follows easily by induction that x_k and y_k are normally distributed.

Suppose that at time $k-1$ we have an estimate $\hat{x}_{k-1}$ of x_{k-1} and an estimate of the covariance P_{k-1}^{xx-} of x_{k-1} with $\hat{x}_{k-1}$ as mean, and also we know $y_{1:k-1}$. We then construct a prior estimate (prediction) of x_k, x_k^-, and calculate the covariance of random variable x_k, P_k^{xx-}, with x_k^- as mean and also predict the value of y_k, y_k^- and calculate the covariance of random variable y_k with mean y_k^-.

When the value of y_k becomes available at time k we construct an updated estimate of random variable $x_k, \hat{x}_k$, by adding the Kalman gain times the $y_k - y_k^-$ to the prior estimate x_k^-. We also calculate the covariance P_k^{xx} of random variable x_k with $\hat{x}_k$ as mean. We also know $y_{1:k}$. We are in a position to start the above process at time k to do estimates for time $k+1$.

The above calculations imply the following with regard to the distributions, where the true means of the variables x_k, y_k are replaced by their estimates, so these represent the estimated distributions, conditional on their means being given as above, and not the true distributions:

$$\begin{aligned} x_k|I_{k-1} &\sim N(x_k^-, P_k^-) \\ &\sim N(A\hat{x}_{k-1} + C_k, AP_{k-1}A^T + Q) \\ x_k|y_k, I_{k-1} &\sim N(\hat{x}_k, P_k) \\ y_k|I_{k-1} &\sim N(y_k^-, F_k) = N(y_k^-, P_k^{yy-}) \end{aligned}$$

So if p denotes probability, we then have

$$\begin{aligned} p(x_k|I_{k-1}) &= \texttt{normpdf}(x_k; A\hat{x}_{k-1} + C_k, AP_{k-1}A^T + Q) \\ p(x_k|y_k, I_{k-1}) &= \texttt{normpdf}(x_k; \hat{x}_k, P_k^{xx}) \\ p(y_k|y_k, I_{k-1}) &= \texttt{normpdf}(y_k; \hat{y}_k^-, P_k^{yy-}) \end{aligned}$$

Theorem A.1. .

Prior Estimates

$$\begin{aligned} \hat{x}_k^- &= A\hat{x}_{k-1} + C_k \\ P_k^- &= AP_{k-1}A^T + Q(= P_k^{xx-}) \end{aligned}$$

Posterior Estimation Minimising the Variance of the e_k

$$\begin{aligned} K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + R)^{-1} = P_k^- H_k^T (F_k)^{-1} (= P_k^{xy-} (P_k^{yy-})^{-1}) \\ \hat{x}_k &= \hat{x}_k^- + K_k(y_k - H_k\hat{x}_k^- - D_k) \\ P_k &= (I - K_k H_k)P_k^- (= P_k^{xx}) \\ &(= P_k^{xx-} - K_k P_k^{yy-} K_k^T) \end{aligned}$$

Algorithm
Choose $\hat{x}_0, P_0$. Then for each time step $k \geq 1$, calculate the Prior estimate for time step k followed by the Posterior estimate for time step k, and iterate for all k.

Proof. The sum of the variances of the components is the trace (sum of the diagonal elements) of the covariance matrix (the diagonal elements are the variances of the components). To minimise $P_k = cov(e_k)$ with respect to K_k we differentiate $tr(P_k)$ wrt each element of K_k and set them equal to zero. We can arrange these differentiated elements as a square matrix $\bigtriangledown_K(tr(P_k))$ by setting $(\bigtriangledown_K(tr(P_k)))_{ij} = \frac{\partial}{\partial(K_k)_{ij}}(tr(P_k))$. This leads to the following rules:

$$\bigtriangledown_K(tr(KA)) = A^T, \quad \bigtriangledown_K(tr(AK^T)) = A, \quad \bigtriangledown_K(tr(KAK^T)) = KA + KA^T$$

If A is symmetric then

$$\bigtriangledown_K(tr(KAK^T)) = 2KA$$

Applying this to the Josephson Form (see above) for P_k which is given by

$$\begin{aligned} P_k &= L_k P_k^- L_k^T + K_k R K_k^T \\ &= P_k^- - K_k(H_k P_k^-) - (P_k^- H^T)K_k^T + K_k(H_k P_k^- H_k^T + R)K_k^T \end{aligned}$$

we get

$$\bigtriangledown_K tr(P_k) = -(H_k P_k^-)^T - P_k^- H_k^T + 2K(H_k P_k^- H_k^T + R)$$

Setting this to zero, recalling that P_k^- is symmetric, as is the last bracketed term, and solving for K_k yields the desired results.

It remains to prove the last equation for P_k. Substituting the optimal value of K_k into the the Josephson Form for P_k yields:

$$\begin{aligned} P_k &= L_k P_k^- - P_k^- H_k^T K^T + K(\cdot)K^T = L_k P_k^- - P_k^- H_k^T K^T + P_k^- H_k^T (\cdot)^{-1}(\cdot)K^T \\ &= L_k P_k^- \end{aligned}$$

□

Note:

1. Instead of the 5th equation in the Theorem we could use the Josephson Form, which is valid whether or not K_k is optimal:

$$\begin{aligned} L_k &= I - K_k H_k \\ P_k &= L_k P_k^- L_k^T + K_k R K_k^T \end{aligned}$$

This form can be more stable although it is more expensive to calculate.

2. If we let $f(x) = Ax + C$, $h(x) = Hx + D$ then we may write the Kalman algorithm in terms of covariance matrices as

$$\begin{aligned}
\hat{x}_k^- &= f(\hat{x}_{k-1}) \\
y_k^- &= h(x_k^-) \\
P_k^{xx-} &= E(e_k^- e_k^{-T} | I_{k-1}) = cov(e_k) (= AP_{k-1}^{xx} A^T + Q) \\
P_k^{xy-} &= E(e_k^- e_k^{y-T} | I_{k-1}) = cov(e_k, e_k^{y-}) (= P_k^{xx-} H^T) \\
P_k^{yy-} &= E(e_k^{y-} e_k^{y-T} | I_{k-1}) = cov(e_k^{y-}) (= HP_k^{xx-} H^T + R) \\
K_k &= P_k^{xy-} (P_k^{yy-})^{-1} \\
\hat{x}_k &= \hat{x}_k^- + K_k (y_k - y_k^-) \\
P_k^{xx} &= (I - K_k H_k) P_k^{xx-}
\end{aligned}$$

3. If K_k is not the (optimal) Kalman gain then the third last equation must be omitted and the last equation must be replaced by the Josephson form.

Appendix B

MATLAB codes

This appendix includes the MATLAB codes. As stated before, we have used $n-$point Gauss Legendre Quadratures for integration. The scrips for Gauss Legendre Quadratures are taken from (Rouah (2013), chapter 2 and chapter 12).

B.1 Heston EKF

```
function LML=EKFH(param,MktPrice,K,T,S,r)
% Extendend Kalman Filter(EKF) for Heston model

% This function uses EKF to estimate implied volatilities via the
% Heston model using put options as the observation.
% This was combined with Maximum Likelihood estimation method to
% evaluate the likelihood function for parameters estimation.

% The data is put option prices on Jow Jones Industrial average (DJIA)
% recorded on May 10, 2012 and we got these data
% from Fabrice Douglas Rouah website in his MATLAB codes named:
% (The Heston model and its extensions in Matlab and C##), in chapter 9,
% script: Mikhailov_Nogel_estimation_DJIA.

% Inputs:
    % K is the strike price
    % S is the stock price
    % T is maturity time
    % rf is the risk free rate
    % q is a dividend
    % kappa=param(1) is the volatility mean reversion rate
    % theta=param(2) is the the volatility mean reversion level
    % sigma=param(3) is the volatility for the volatility
    % v00=param(4) is the initial volatility
    % rho=param(5) is the correlation
    % MktPrice is the market put prices, where the market data was
```

```
    % generated from the quoted implied vols using the Black-Scholes
    % formula.

    % Unknown
    % v0= the states transition equation (Heston model volatility process)
    % v0=v0+kappa*(theta-v0)*dt+noise;

    % ModelPrice=observations, which is the put prices of Heston model
    % ModelPrice= CallPrice - S0 + exp(-rf*T(t))*K(k), via put-call parity.

rf=r;
S0=S;
[NK,NT] = size(MktPrice);

% X=zeros(NK,NT);  % matrix for the heston volatility at each K and T
v=zeros(NK,NT);
ModelPrice=zeros(NK,NT);
% LML=0.5*NK*log(2*pi);

% Read in the parameter estimates and weights and abscissas

% Hard code the parameter estimates
kappa = param(1);
theta = param(2);
sigma = param(3);
v00   = param(4);
rho   = param(5);
lambda = 0;
LML=0;
% mu=r;
dt=1/252;
P=0.863; % initial covariance
% Q=sigma*v00*dt; % covariance matrix for the process noises
R=0.00005536; % covariance matrix for the measurement noises
trap = 1;
% trap: 1 = "Little Trap" formulation for the Heston characteristic function
% or the "Little Heston Trap" formulation of Albrecher et al.
%        0 = Original Heston formulation
% Weights and abscissas
% x = Gauss Laguerre abscissas
% w = Gauss Laguerre weights
[x,w] = GenerateGaussLaguerre(32);
% EKF, where we have two steps, the prediction and update step.
v0=v00;
for t=1:NT
    for k=1:NK
        % prediction step
        v0=v0+kappa*(theta-v0)*dt; % volatility prediction (our state)

        % performing Jacobian
        A=1-kappa*dt; % 1st derivative for the transtion
                      %wrt states
```

```
%          Q=(v0-theta)*((1-exp(-kappa*dt))/kappa)+theta*dt;
        W=sigma*sqrt(v0)*sqrt(dt); % jacobian matrix for the process noise
        Q=W;
        P=A*P*A'+W*Q*W'; % covariance for the predicted states

        % Update step
        % First find the Heston put prices, our measurement
        % We use function HestonCallGaussLaguerre from chapter 2, Fabrice
        % Douglas Rouah matlab code, please add to path chapter 2.

        % By put-call parity
        ModelPrice(k,t)= HestonCallGaussLaguerre(S0,K(k),T(t),rf,kappa,...
            theta,sigma,lambda,v0,rho,trap,x,w)- S0 + exp(-rf*T(t))*K(k);

        % central-difference method for finding the option 1st derivative
        % wrt the state
        dv = 1e-4;
        V1 = HestonCallGaussLaguerre(S0,K(k),T(t),rf,kappa,theta,sigma,...
            lambda,v0+dv,rho,trap,x,w)- S0 + exp(-rf*T(t))*K(k);
        V2 =  HestonCallGaussLaguerre(S0,K(k),T(t),rf,kappa,theta,sigma,...
            lambda,v0-dv,rho,trap,x,w)- S0 + exp(-rf*T(t))*K(k);
        H = (V1-V2)/2/dv*2*sqrt(v0); % first derivative for the observation
                                     % wrt the state values

        U=dv*sqrt(dt); % Jacobian for the option noise

        %  Measurement residual
        v(k,t)=ModelPrice(k,t)-H*v0;

        F=H*P*H'+U*R*U'; % Measurement covariance
        Finv=1/F;
        Kr=P*H'*Finv; % Kalman gain matrix

        v0=abs(v0+Kr*v(k,t)); % new state estimate
        P=P-Kr*H*P; % updated covariance for the new estimate

        LML=LML+log(F)+v(k,t)*Finv; % likelihood function via MLE

    end
end
```

B.2 Heston UKF

```
function LM=Heston_UKF(MktPrice,param,alpha,ki,T,K,S,r,beta,P)
% Unscented Kalman Filter (UKF)

% This function uses UKF to estimate implied volatilities of
% the Heston model. We have additive process and measurement noise.
% This was combined with Maximum Likelihood estimation method to
```

```
% evaluate the likelihood function for parameters estimation.

% Observation:
        % put Options
% Unobservable State :
        % variance
% To combute Heston characteristic function we use
% Gauss-Laguerre Quadrature


rf=r;
S0=S;

[NK,NT] = size(MktPrice);
% set of parameters
kappa = param(1);
theta = param(2);
sigma = param(3);
v00   = param(4);
rho   = param(5);

lambda = 0;
dt=1/252;

trap = 1;

[x,w] = GenerateGaussLaguerre(32);

Q=sigma*v00*dt; % covariance for the process noise
R=1e-6; % covariance for the measurement noise

%Augmented state vector
xx=0;%zeros(2,1);
% Pz=0;
% Pxz=0;%zeros(2,1);
% z=zeros(NK,NT);
z=0;
LM=0;
ModelPrice=zeros(NK,NT);

for k=1:NK
    for t=1:NT

        Vpred=v00;
        Pa=P;
        % Augmented mean
        Xa=Vpred;
        % Augmented matrix covariance

        na=length(Xa);
        m=2*na;
```

```
        % Sigma points
%         alpha=1e-3;%0.99; %alpha is recommended to be between 10e-3 and 1
%         ki=0;
%         beta=2;
        lambdaa=alpha*alpha*(na+ki)-na;

        PO=sqrt((na+lambdaa)*real(Pa));

        Xb=Xa;

        Wm=zeros(1,m+1);
        Wc=zeros(1,m+1);
        for i=1:m+1

            % Weights
            if i~=1
                Wm(i)=1/(2*(na+lambdaa));
                Wc(i)=1/(2*(na+lambdaa));

            else
                Wm(i)=lambdaa/(na+lambdaa);
                Wc(i)=(lambdaa/(na+lambdaa))+(1-alpha*alpha+beta);

            end
         %{
            if i==na
                Xc(i)=Xa+PO;
            else
                Xc(i)=Xa-PO;
            end
%}

            Xi=[Xb Xa+PO Xa-PO]; % set of sigma points
    %         f=zeros(2,m+1);
        % Prediction step
            f=Xi(1,i)+kappa*(theta-Xi(1,i))*dt;%+Xi(2,i);  %propagate sigma points

            xx=xx+Wm(i)*f; % estimated mean
            P=((f-xx)*Wc(i)*(f-xx)) + Q;   % prior covariance

        % Update step

% Put option prices via Gauss-Laguerre Quadrature
            ModelPrice(k,t) = HestonCallGaussLaguerre(S0,K(k),T(t),rf,kappa,...
                theta,sigma,lambda,xx,rho,trap,x,w) - S0 + exp(-rf*T(t))*K(k);% put
                                                        % put-call parity.

            z=z+Wm(i)*ModelPrice(k,t);
            zz=ModelPrice(k,t)-z;
            Pz=zz*Wc(i)*zz+R;
            Pxz=(f-xx)*Wc(i)*zz;
        end
```

```
        % Kalman gain
        Kr=Pxz/Pz;

        % new estimate
        v00=abs(xx+Kr*(zz));
        P=P-Kr*Pz*Kr;

        loglik=log(Pz)+(MktPrice(k,t)-z)/Pz;

%         loglik=(-NK/2)*log(2*pi)-0.5*log(Pz)-0.5*(MktPrice(k,t)-z)*(MktPrice(k,t)

        LM=LM+loglik; % likelihood function

    end
end
```

B.3 Heston PF

```
function LM = Heston_PF(MktPrice,param,N,K,T,S,r)
% Particle Filtering (PF)
% This function uses PF to estimate implied volatilities of
% the Heston model.
% This was combined with Maximum Likelihood estimation method to
% evaluate the likelihood function for parameters estimation.

% Unobservable State :
        % variance
% To combute Heston characteristic function, we use
% Gauss-Laguerre Quadrature
% N=Number of Particles

    kappa = param(1);
    theta = param(2);
    sigma = param(3);
    v0 = param(4);
    rho = param(5);


    lambda = 0;

    trap = 1;

    [x,w] = GenerateGaussLaguerre(32);

    rf=r;
    S0=S;
    q  = 0.0068;
    dt=1/252;
```

```
    [NK,NT] = size(MktPrice);

    LM=0;

    xx=v0;

% Particle filter
     X = xx *ones(1,N);

for j=1:N
    for k=1:NK
        for t=1:NT

            xPred=xx;
            Q = (theta*sqrt(1-rho^2)*sqrt(xPred*dt))^2 + 1e-9;

            % Sampling step
            X=X+sqrtm(Q)*randn(1,N);

            % Prediction step
            X = X+kappa*(theta-X)*dt;
            X=abs(X);

            % Put option via Gauss-Laguerre Quadrature of Heston
            % Characteristic func.

            ModelPrice(k,t,j) = HestonCallGaussLaguerre(S0,K(k),T(t),rf,kappa,...
                theta,sigma,lambda,X(j),rho,trap,x,w) - S0 + exp(-rf*T(t))*K(k);

            y(:,:,j)=ModelPrice(k,t,j);
            R =  (xx*dt)+ 1e-9;

            ww=exp(-0.5*y(:,:,j)'*inv(R)*y(:,:,j))+1e-15; %particles weight

            ww=ww/sum(ww); % normalize the weights

            yMean=sum(ww.*ModelPrice(k,t),1); % predicted measurements

            xx=sum(ww.*X); % state estimate
            PP=ModelPrice(k,t)-yMean;
            P = (PP*diag(ww)*PP')+R; % measurement  Cov

            % Measurement  Residual
            MR = ModelPrice(k,t)-yMean;
            % Likelihood
            LM=LM+0.5*log(det(P))+0.5*MR'*inv(P)*MR;

            % Resampling step
            NewIndex = Resample(ww);

            X(:)=X(NewIndex);
```

```
        end
    end
end
end


function NewIndex = Resample(ww)
% This function Resample(ww) is taken from Mastro book, Chap: 9
% Resample resamples [OldIndex=1:NumParticles; OldWeights=w]
% to [NewIndex ; NewWeights=1/NumPart.,...,1/NumPart.]
% e.g., NewIndex = [1 1 4 5 5 6 6 6 6]
% Excellent comparison of resampling schemes available from
% Nando de Freitas (Berkeley) and Rudolph van der Merwe (OGI)
    L=max(size(ww));
    OldIndex=1:L;
    NewPart=zeros(1,L);
    CDFunc=cumsum(ww);

    u = sort(rand(1,L));
    % Nando de Freitas and Rudolph van der Merwe suggest
    % u=fliplr(cumprod(rand(1,L).^(1./(L:-1:1))));
    j=1;
    for counter=1:L
        while(u(counter) > CDFunc(j))
            j=j+1;
        end
        NewPart(j)=NewPart(j)+1;
    end

    ind=1;
    for counter=1:L
        if (NewPart(1,counter)>0)
            for j=ind:ind+NewPart(counter)-1
                NewIndex(j) = OldIndex(counter);
            end
        end
        ind=ind+NewPart(counter);
    end
end
```

B.4 Double Heston EKF

```
function likelihood=EKFDH(MktPrice,param,K,T,S,r)
% Extendend Kalman Filter(EKF) for the Double Heston model
% This function uses EKF to estimate implied volatilities of the
% Double Heston model.
% This was combined with Maximum Likelihood estimation method to
% evaluate the likelihood function for parameters estimation.

% Inputs:
```

```
    % K is the strike price
    % S is the stock price
    % T is maturity time
    % rf is the risk-free rate
    % q  = dividend yield-rate
    % param = Two sets of Double Heston parameters
    %         [kappa1 theta1 sigma1 v01 rho1,kappa2 theta2 sigma2 v02 rho2]
    % V01 and V02= the state transition equations
    % V01=v01+kappa1*(theta1-v01)*dt+noise
    % V02=v02+kappa2*(theta2-v02)*dt+noise;

    % ModelPrice=observations, which is the put prices from double Heston
    % ModelPrice=DoubleHestonPriceGaussLaguerre(P,S,K(k),T(t),rf,q,param,x,w,trap);

rf=r;
q  = 0.0068;
PutCall = 'P';
[NK,NT] = size(MktPrice);

likelihood=0;

% Read in the weights and abscissas
[x,w] = GenerateGaussLaguerre(32);

kappa1 = param(1); theta1 = param(2); sigma1 = param(3); v01= param(4);
rho1 = param(5);
kappa2 = param(6); theta2 = param(7); sigma2 = param(8); v02= param(9);
rho2 = param(10);


dt=1/252;
P1=0.03; % initial covariance
P2=0.02;
v00=[v01 v02]';
R=0.005536; % covariance matrix for the measurement noises
ModelPrice=zeros(NK,NT);
v=zeros(NK,NT);
% "Little Trap, trap=1" formulation for the Heston characteristic function
trap = 1;
for t=1:NT %NT=4

    for k=1:NK  %NK=13
        Vpred=v00;
        if k==1
            P=[P1 0; 0 P2];
        else
            P=P;
        end

        % prediction step

        V=[Vpred(1,1)+kappa1*(theta1-Vpred(1,1))*dt; Vpred(2,1)+kappa2*...
```

```
    (theta2-Vpred(2,1))*dt]; % prediction of the states
%(v02-theta2)*exp(-kappa2*dt)+theta2; % prediction of the states

% covariance matrix for the process noises
Q=[sigma1*Vpred(1,1)*dt 0;0 sigma2*Vpred(2,1)*dt];

% 1st derivative for the transtion wrt states
A=[1-kappa1*dt 0; 0 1-kappa2*dt];

W=[sigma1*sqrt(Vpred(1,1))*sqrt(dt) 0; 0 sigma2*sqrt(Vpred(2,1))*...
    sqrt(dt)]; % jacobian matrix for the process noise

P=A*P*A'+W*Q*W'; % covariance for the predicted process

% observations prices
% this function was borrowed from chapter 12 in Fabrice
% Douglas Rouah matlab code, please add to path chapter 12
param(4)=Vpred(1,1);  % assign new value of v01
param(9)=Vpred(2,1);  % assign new value of v02

ModelPrice(k,t) = DoubleHestonPriceGaussLaguerre(PutCall,S,K(k),...
    T(t),rf,q,param,x,w,trap);

% Perform Jacobian
% central-difference method
dv = 1e-2;
param(4)=Vpred(1,1)+dv;
V01 = DoubleHestonPriceGaussLaguerre(PutCall,S,K(k),T(t),rf,q,param....
    ,x,w,trap);
param(4)=Vpred(1,1)-2*dv;
V02 = DoubleHestonPriceGaussLaguerre(PutCall,S,K(k),T(t),rf,q,param....
    ,x,w,trap);

param(4)=Vpred(1,1)+dv;

param(9)=Vpred(2,1)+dv;
V11=DoubleHestonPriceGaussLaguerre(PutCall,S,K(k),T(t),rf,q,param,...
    x,w,trap);
param(9)=Vpred(2,1)-2*dv;
V12 = DoubleHestonPriceGaussLaguerre(PutCall,S,K(k),T(t),rf,q,...
    param,x,w,trap);

param(9)=Vpred(2,1)+dv;

% first derivative for the observation wrt the state values
H1 = (V01-V02)/2/dv*2*sqrt(Vpred(1,1));
H2 = (V11-V12)/2/dv*2*sqrt(Vpred(2,1));
H=[H1 H2];
U=[dv*sqrt(dt) dv*sqrt(dt)];% Jacobian for the option noise, which
                    %is assumed to be the same for both v01 and v02

% update step
```

```
        % Measurement residual
        v(k,t)=ModelPrice(k,t)-H*V;

        F=H*P*H'+U.*R*U'; % Measurement covariance

        Finv=inv(F);

        Kr=P*H'.*Finv; % Kalman gain matrix

        v00=abs(V+Kr.*v(k,t)); % new states

        P=P-Kr*H*P; % updated covariance for the new estimated states

        % Likelihood function
        likelihood=likelihood+log(F)+v(k,t)*Finv;
    end

end
```

B.5 Double Heston UKF

```
function LM=Double_UKF(MktPrice,param,alpha,ki,T,K,S,r,beta)

% This function uses UKF to estimate implied volatilities of
% the Double Heston model. There is additive process and measurement noise.
% This was combined with Maximum Likelihood estimation method to
% evaluate the likelihood function for parameters estimation.
% Observation:
        % put Options
% Unobservable State :
        % variance
% To compute double Heston characteristic function, we use
% Gauss-Laguerre Quadrature

rf=r;
S0=S;
q  = 0.0068;
PutCall = 'P';
[NK,NT] = size(MktPrice);
% set of parameters
kappa1 = param(1); theta1 = param(2); sigma1 = param(3); v01= param(4);
rho1 = param(5);
kappa2 = param(6); theta2 = param(7); sigma2 = param(8); v02= param(9);
rho2 = param(10);

dt=1/252;

trap = 1;
```

```
[x,w] = GenerateGaussLaguerre(32);

%Augmented state vector
xx=zeros(2,1);
v00=[v01 v02]';
% Pz=0;
% Pxz=0;%zeros(2,1);
% z=zeros(NK,NT);
z=0;
LM=0;
ModelPrice=zeros(NK,NT);

for k=1:NK
    for t=1:NT

        Vpred=v00;
        if t==1
            Pa=[rho1 0; 1 rho2];
        else
            Pa=P;
        end

        % Augmented mean
        Xa=Vpred;
        % Augmented matrix covariance

        na=length(Xa); % 2x1
        m=2*na;        % 4

        % Sigma points
%         alpha=1e-3;%0.99; %alpha is recommended to be between 10e-3 and 1
%         ki=0;
%         beta=2;
        lambdaa=alpha*alpha*(na+ki)-na;

        PO=sqrt((na+lambdaa).*real(Pa));

        Xb=Xa;

        Wm=zeros(1,m+1);
        Wc=zeros(1,m+1);
        for i=1:m+1

            % Weights
            if i~=1
                Wm(i)=1/(2*(na+lambdaa));
                Wc(i)=1/(2*(na+lambdaa));

            else
                Wm(i)=lambdaa/(na+lambdaa);
                Wc(i)=(lambdaa/(na+lambdaa))+(1-alpha*alpha+beta);
```

```
            end

            Xc=zeros(na,2*na);
            for j=1:na
                Xc(:,j)=Xa+PO(:,j);
            end
            for j=na+1:m
                Xc(:,j)=Xa-PO(:,j-na);
            end

            Xi=[Xb Xc]; % set of sigma points

    %         f=zeros(2,m+1);
        % Prediction step
            f=[Xi(1,i)+kappa1*(theta1-Xi(1,i))*dt;Xi(2,i)+kappa2*...
                (theta2-Xi(2,i))*dt];

            xx=xx+Wm(i).*f; % predicted states
            P=Wc(i).*(f-xx)*(f-xx)' ;   % predicted state covariance

            param(4)=xx(1,1);
            param(9)=xx(2,1);


% Put option prices via Gauss-Laguerre Quadrature
            ModelPrice(k,t) = DoubleHestonPriceGaussLaguerre(PutCall,S,K(k),T(t),rf

            z=z+Wm(i)*ModelPrice(k,t); % prediction of observations
            zz=ModelPrice(k,t)-z;
            Pz=zz*Wc(i)*zz; % predicted observations covariance
            Pxz=(f-xx)*Wc(i)*zz; % joint covarinace of state and observation
        end

         % Update step

        % Kalman gain
        Kr=Pxz/Pz; % 2x1

        % new state estimate
        v00=abs(xx+Kr.*(zz));
        P=P-Kr*Pz*Kr'; % state covariance

        loglik=log(Pz)+(MktPrice(k,t)-z)/Pz;

%loglik=(-NK/2)*log(2*pi)-0.5*log(Pz)-0.5*(MktPrice(k,t)-z)*(MktPrice(k,t)-z)/Pz;

        LM=LM+loglik; % likelihood function

    end
end
```

B.6 Double Heston PF

```
function LM = Double_PF(MktPrice,param,N,K,T,S,r)
% Particle Filtering for the Double Heston model

% This function uses PF to estimate implied volatilities of
% the Double Heston model. We have additive process and measurement noise.
% This was combined with Maximum Likelihood estimation method to
% evaluate the likelihood function for parameters estimation.

% Observation:
        % put Options
% Unobservable State :
        % variance
% To compute double Heston characteristic function, we use
% Gauss-Laguerre Quadrature
% N=Number of Particles

kappa1 = param(1); theta1 = param(2); sigma1 = param(3);
v01= param(4); rho1 = param(5); kappa2 = param(6);
theta2 = param(7); sigma2 = param(8); v02= param(9);
rho2 = param(10);


    lambda = 0;

    trap = 1;

    [x,w] = GenerateGaussLaguerre(32);


 % Initialize Process Covariance
    Pxx=[rho1 0; 1 rho2];

    rf=r;
    S0=S;
    q  = 0.0068;
    dt=1/252;
    [NK,NT] = size(MktPrice);
    PutCall = 'P';
    LM=0; %0.5*nSteps*log(2*pi);

    xx=[v01 v02]';

     X = xx *ones(1,N);

for j=1:N
    for k=1:NK
        for t=1:NT

            xPred=xx;
```

```
            Q=[sigma1*xPred(1,1)*dt 0;0 sigma2*xPred(2,1)*dt];

            % Sampling step
            X=X+sqrtm(Q)*randn(2,N);

            % Prediction step
            X = X+([kappa1*(theta1-X(1,:)); kappa2*(theta2-X(2,:))])*dt;
            X=abs(X);

            % Put option via Gauss-Laguerre Quadrature of Heston Characteristic fun

            param(4)=X(1,j);
            param(9)=X(2,j);

            ModelPrice(k,t,j) = DoubleHestonPriceGaussLaguerre(PutCall,...
                S,K(k),T(t),rf,q,param,x,w,trap);

            y(:,:,j)=ModelPrice(k,t,j);
            R =  diag(xPred*sqrt(dt))';

            % particle weight
            ww=exp(-0.5*y(:,:,j)'*inv(R)*y(:,:,j))+1e-15;

            ww=ww/sum(ww); % normalize the weights

            yMean=sum(ww.*ModelPrice(k,t),1);

            xx=[sum(ww(1,1).*X(1,j)) sum(ww(2,1).*X(2,j))]';
            PP=ModelPrice(k,t)-yMean;
            P= (PP*diag(ww)*PP')+R; % measurement covariance

            % Measurement  Residual
            MR = ModelPrice(k,t)-yMean; % option price error
            % Likelihood function
            LM=LM+sum(sum(0.5*log(det(Pyy))+0.5*MR'*inv(Pyy)*MR));

            % Resampling step
            NewIndex = Resample(ww);

            X(:,j)=X(NewIndex);

        end
    end
end
end
```

List of References

Albrecher, H., Mayer, P., Schoutens, W. and Tistaert, J. (2006). The little heston trap.

Albrecher, H., Mayer, P. and Schoutens, W.T. (2007). The little heston trap. *Wilmott Magazine, January issue*, pp. 83–92.

Andersen, T.G., Benzoni, L. and Lund, J. (2002). An empirical investigation of continuous-time equity return models. *The Journal of Finance*, vol. 57, no. 3, pp. 1239–1284.

Andersen, T.G. and Bollerslev, T. (1997). Heterogeneous information arrivals and return volatility dynamics: Uncovering the long-run in high frequency returns. *The Journal of Finance*, vol. 52, no. 3, pp. 975–1005.

Arulampalam, M.S., Maskell, S., Gordon, N. and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *Signal Processing, IEEE Transactions on*, vol. 50, no. 2, pp. 174–188.

Bakshi, G., Cao, C. and Chen, Z. (1997). Empirical performance of alternative option pricing models. *The Journal of Finance*, vol. 52, no. 5, pp. 2003–2049.

Ball, C.A. and Roma, A. (1994). Stochastic volatility option pricing. *Journal of Financial and Quantitative Analysis*, vol. 29, no. 04, pp. 589–607.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, vol. 31, no. 3, pp. 307–327.

Chernov, M. and Ghysels, E. (2000). A study towards a unified approach to the joint estimation of objective and risk neutral measures for the purpose of options valuation. *Journal of financial economics*, vol. 56, no. 3, pp. 407–458.

Christoffersen, P., Heston, S. and Jacobs, K. (2009). The shape and term structure of the index option smirk: Why multifactor stochastic volatility models work so well. *Management Science*, vol. 55, no. 12, pp. 1914–1932.

Christoffersen, P. and Jacobs, K. (2004). The importance of the loss function in option valuation. *Journal of Financial Economics*, vol. 72, no. 2, pp. 291–318.

Christoffersen, P., Jacobs, K. and Mimouni, K. (2010). Volatility dynamics for the s&p500: evidence from realized volatility, daily returns, and option prices. *Review of Financial Studies*, vol. 23, no. 8, pp. 3141–3189.

Costabile, M., Massabo, I. and Russo, E. (2012). On pricing contingent claims under the double heston model. *International Journal of Theoretical and Applied Finance*, vol. 15, no. 05.

Doucet, A., De Freitas, N., Murphy, K. and Russell, S. (2000). Rao-blackwellised particle filtering for dynamic bayesian networks. In: *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pp. 176–183. Morgan Kaufmann Publishers Inc.

Duffie, D., Pan, J. and Singleton, K. (2000). Transform analysis and asset pricing for affine jump-diffusions. *Econometrica*, vol. 68, no. 6, pp. 1343–1376.

Engle, R.F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica: Journal of the Econometric Society*, pp. 987–1007.

Gouriéroux, C. (2012). *ARCH models and financial applications*. Springer Science & Business Media.

Guo, B., Han, Q. and Zhao, B. (2014). The nelson–siegel model of the term structure of option implied volatility and volatility components. *Journal of Futures Markets*, vol. 34, no. 8, pp. 788–806.

Hermann, R. and Krener, A.J. (1977). Nonlinear controllability and observability. *IEEE Transactions on automatic control*, vol. 22, no. 5, pp. 728–740.

Heston, S.L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review of Financial Studies*, vol. 6, no. 2, pp. 327–343.

Hol, J.D., Schon, T.B. and Gustafsson, F. (2006). On resampling algorithms for particle filters. In: *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*, pp. 79–82. IEEE.

Hull, J. and White, A. (1987). The pricing of options on assets with stochastic volatilities. *The Journal of Finance*, vol. 42, no. 2, pp. 281–300.

Hurn, A.S., Lindsay, K.A. and McClelland, A.J. (2015). Estimating the parameters of stochastic volatility models using option price data. *Journal of Business & Economic Statistics*, vol. 33, no. 4, pp. 579–594.

Javaheri, A. (2011). *Inside volatility arbitrage: the secrets of skewness*, vol. 317. John Wiley & Sons.

Javaheri, A., Lautier, D. and Galli, A. (2003). Filtering in finance. *Wilmott*, vol. 3, pp. 67–83.

Julier, S.J. and Uhlmann, J.K. (1996). A general method for approximating nonlinear transformations of probability distributions. Tech. Rep., Technical Report, RRG, Department of Engineering Science, University of Oxford.

Kalman, R.E. (1960). A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, vol. 82, no. 1, pp. 35–45.

Kienitz, J. and Wetterau, D. (2012). *Financial modelling: Theory, implementation and practice with matlab source*. John Wiley & Sons.

Lagarias, J.C., Reeds, J.A., Wright, M.H. and Wright, P.E. (1998). Convergence properties of the nelder–mead simplex method in low dimensions. *SIAM Journal on optimization*, vol. 9, no. 1, pp. 112–147.

Li, J. (2013). An unscented kalman smoother for volatility extraction: Evidence from stock prices and options. *Computational Statistics & Data Analysis*, vol. 58, pp. 15–26.

Lu, L., Yang, M., Geng, S., Wang, S.-f. and Wang, Y.-h. (2015). Study on importance function for particle filter. *International Journal of Multimedia and Ubiquitous Engineering*, vol. 10, no. 2, pp. 249–258.

Nason, J.M. and Smith, G. (2013). Reverse kalman filtering us inflation with sticky professional forecasts.

Pan, J. (2002). The jump-risk premia implicit in options: Evidence from an integrated time-series study. *Journal of financial economics*, vol. 63, no. 1, pp. 3–50.

Reif, K., Günther, S., Yaz, E. and Unbehauen, R. (1999). Stochastic stability of the discrete-time extended kalman filter. *IEEE Transactions on Automatic Control*, vol. 44, no. 4, pp. 714–728.

Rouah, F.D. (2013). *The Heston model and its extensions in MATLAB and C#*. John Wiley & Sons.

Rouah, F.D. (2013 (accessed September 3, 2015)). The heston model and its extensions in matlab and c#.
Available at: `http://eu.wiley.com/WileyCDA/Section/id-818210.html`

Scott, L.O. (1987). Option pricing when the variance changes randomly: Theory, estimation, and an application. *Journal of Financial and Quantitative Analysis*, vol. 22, no. 04, pp. 419–438.

Shephard, N. (2005). Stochastic volatility. *Economics Group, Nuffield College, University of Oxford, Economics Papers*.

Sira-Ramirez, H. (1988). Algebraic condition for observability of non-linear analytic systems. *International Journal of Systems Science*, vol. 19, no. 11, pp. 2147–2155.

Stein, E.M. and Stein, J.C. (1991). Stock price distributions with stochastic volatility: an analytic approach. *Review of financial Studies*, vol. 4, no. 4, pp. 727–752.

Tanizaki, H. (2013). *Nonlinear filters: estimation and applications*. Springer Science & Business Media.

Théoret, R. and Racicot, F.-É. (2010). Forecasting stochastic volatility using the kalman filter: An application to canadian interest rates and price-earnings ratio.

Thrun, S. (2002). Particle filters in robotics. In: *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pp. 511–518. Morgan Kaufmann Publishers Inc.

Van Der Merwe, R., Doucet, A., De Freitas, N. and Wan, E. (2000). The unscented particle filter. In: *NIPS*, vol. 2000, pp. 584–590.

Wan, E.A. and Van Der Merwe, R. (2000). The unscented kalman filter for nonlinear estimation. In: *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pp. 153–158. Ieee.

Zhu, J. (2009). *Applications of Fourier transform to smile modeling: Theory and implementation*. Springer Science & Business Media.